Cahiers GUTenberg

INTRODUCTION À SGML, DSSSL ET SPDL

Michel Goossens, Eric Van Herwijnen

Cahiers GUTenberg, nº 12 (1991), p. 37-56.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1991___12_37_0>

Introduction à SGML, DSSSL et SPDL*

Michel Goossens et Eric van Herwijnen

CERN, CH-1211 Genève 23, Suisse

Résumé

Cet article passe en revue quelques aspects de la norme ISO 8879 SGML, le langage normalisé de balisage généralisé, et la situe par rapport à d'autres projets de normes pour la description des documents sur ordinateur : DSSSL pour la mise en page et SPDL pour la visualisation.

Mots clés

Traitement de texte ; SGML; DSSSL; SPDL; PostScript ; standards ISO

Abstract

This article provides an introduction to ISO Standard 8879 SGML, the "Standard Generalized Markup Language" and discusses its relation with two other standards being drafted in the area of electronic document description, DSSSL for the page layout and SPDL for the visual presentation

1. Interêt de la norme SGML

SGML — langage normalisé de balisage généralisé ou *Standard Generalised Markup Language* en anglais — est une norme internationale ISO (Organisation internationale de normalisation dont le siège est à Genève) qui définit un formalisme de spécification pour le **balisage** de documents sous forme électronique afin de faciliter les échanges entre des systèmes différents de traitement de texte ou d'information. Comme la norme le précise [1], SGML « peut être utilisé dans l'édition, au sens le plus général de ce terme, qui s'étend de l'édition traditionelle sur support unique jusqu'à l'édition multimédia à partir de bases de données. SGML peut également être utilisé pour traiter les documents en environnement bureautique, lorsque l'on

*Cet article est l'adaptation française de la première partie de l'article "Scientific Text Processing" qui doit paraître dans *International Journal of Modern Physics C.* Une première version de cet article en français est parue dans le numéro spécial sur les échanges du *Flash Informatique* de l'EPFL, été 1991, sous le titre : "Une introduction à SGML et sa relation avec DSSSL et SPDL", p. 17–24.

recherche à la fois la possibilité de relecture par des humains et de transfert vers des systèmes de publication. »

On peut se rendre compte de l'importance de SGML pour l'édition française en sachant que c'est une des rares normes à avoir été traduite en français. Pour une introduction pratique à SGML le lecteur peut consulter [2] ou [3].

Historiquement, le balisage était le processus par lequel un rédacteur introduisait des marques (balises) dans un manuscrit pour indiquer à l'imprimeur comment il fallait composer (formater) le manuscrit. Dans la plupart des cas il s'agissait de commentaires écrits à la main comme par exemple : « Composez le titre avec la police Helvetica-Medium-Italic corps 12 et avec un interlignage de 14 points, justifié sur une largeur de 22 picas avec un renfoncement de un demi cadratin à gauche et de zéro à droite. »

Avec l'apparition des ordinateurs, on a pu coder ces commandes à l'aide d'un système d'encodage spécial. Chaque photo-composeuse avait son propre « langage » spécifique et unique, qu'on appelait langage de balisage par analogie avec les vieux systèmes manuels. La figure 1 montre un exemple de balisage utilisant le langage NORTEXT [4] et l'image scannée du résultat obtenu à la sortie:

```
<CC 15,5,12>On demande la police numéro 5, en corps 12 et avec une
justification de 15 unités.
<SS><QL> <CC 20,8>Maintenant, on prend la police 8 et une
justification de 20 unités.
<QL> <RS>Et maintenant on repart avec les valeurs typographiques
initiales.<EP>
```

ce qui donnait après composition :

On demande la police numéro 5,
en corps 12 et avec une justification
de 15 unités.
Maintenant, on prend la police 8 et une justification de 20 unités.
Et maintenant on repart avec les
valeurs typographiques initiales.

Figure 1. Exemple d'un langage de photo-composition (NORTEXT)

Des entreprises spécialisées en photocomposition ou des unités intégrées à l'intérieur des grandes sociétés offraient souvent des services de

« dactylographie ». Le format de balisage des documents à composer sur le système « maison » restait toujours le même et par conséquent le manque de compatibilité entre les différents systèmes n'était pas important.

Cependant lorsque les auteurs et les clients des entreprises de composition commencèrent à saisir leurs documents eux-mêmes, cette situation créa des problèmes. En effet, il n'était possible de faire la saisie et le balisage de son document que si l'on connaissait le format de composition utilisé par la maison d'édition ou de photocomposition. Si l'on utilisait les marques d'un système donné, il y avait de fortes chances que celles-ci soient incompatibles avec celles utilisées par un autre système de composition.

La situation ne s'améliora pas lorsque l'on commença à utiliser des ordinateurs pour préparer les documents. Comme dans le cas des systèmes de photocomposition, les documents étaient balisés avec des commandes **spécifiques** : c'est-à-dire des commandes de bas niveau comme « aller à la ligne », « centrer le texte qui suit », « aller à la page suivante », etc.

Considérons un document contenant les balises spécifiques suivantes :

```
.pa ;.sp 2 ;.ce ;.bd
Titre du chapitre
.sp
```

Il ne peut être converti dans un autre système de composition qu'à un coût élevé. Un autre exemple de document contenant un balisage spécifique est celui montré par la figure 2.

C'est pour améliorer la compatibilité entre les systèmes de balisage qu'a vu le jour un mouvement pour créer un langage de balisage standard. On espérait persuader tous les fournisseurs d'appareils de composition d'accepter un format commun et on considérait que c'était la tâche des entreprises de composition de développer les outils pour traduire ce langage commun en celui de leurs propres machines de photo-composition. Ce langage de balisage commun était de type **générique**. Un balisage générique décrit la fonction **logique** des différents éléments d'un texte dans le contexte de la **structure** du document. Par exemple, on indique à l'aide de balises les paragraphes, les titres, les notes en bas de page. La manière dont un élément sera représenté sur la page n'est pas spécifié dans le texte ce qui donne un certain nombre d'avantages, dont la possibilité de passer d'un format à un autre.

LaTeX est un exemple d'un **langage de balisage générique.** Le tableau 1 montre une comparaison entre plusieurs langages de

```
%PAGE DE TITRE
\banner\bigskip\begingroup\titlefont\obeylines
\vskip 20pt
\hfil TRANSFORMATION DE LA MATIERE ORGANIQUE \hfill
\hfil PAR LA LUMIERE \hfill
\vskip 10pt
\endgroup\bigskip
%
\medskip
\centerline{J.~Cleymans\footnote{${}^{1}$}\SA ,
   { K.~Redlich\footnote{${}^{2}$}\BIL}$^,$\footnote{${}^{3}$}\PO~ et~
H.~$\rm Satz^{2,}$\footnote{${}^{4}$}\CC }
\medskip
%
\bigskip\bigskip\bigskip\bigskip\bigskip\bigskip\bigskip\bigskip
\centerline{{\bf Résumé}}\medskip
L'énergie $E$ transportée par un photon et absorbée par une
molécule est proportionnelle à la fréquence de la radiation
$\nu$, ou inversement proportionnelle à la longueur d'onde
$\lambda$ de cette radiation. Ces paramêtres sont liés
par la relation $e={\rm h}\nu={\rm h c}/\lambda$ où h est la constante
de Planck et c la vitesse de la lumière.
\vfil\eject
\nopagenumbers
\eject\line{}\vfil\eject
\footline={\hss\rm\folio\hss}
\pageno=1
\par\noindent{\bf 1. Introduction}
\vskip 0.25cm
Toute transfomration de matière suppose un changement de
son état énergétique.$^{1-11}$ <etc.>
```

Figure 2. Exemple de balisage spécifique (TEX)

balisage générique (SGML avec la DTD de l'AAP (*American Association of Publishers* : association des éditeurs américains) AAP-Article [5], BookMaster [6] d'IBM, LaTeX [7] et Vax Document [8]. Pour de nombreuses entrées, les différences entre les commandes sont triviales (par exemple `<FN>`, `:FN.`, `\footnote` et `<FOOTNOTE>` sont utilisés pour baliser une note en bas de page). Ceci suggère qu'une conversion d'un système à un autre devrait être possible, à condition toutefois que les langages décrivent la même structure générique.

Parmi les autres avantages du balisage générique, signalons que le document devient indépendant du logiciel de mise en page et que son style est défini à un **seul** endroit, dans une feuille de style ou dans des macros.

Les activités initiales dans le domaine du balisage générique ont été coordonnées par *Graphic Communications Association (GCA)*, un groupe professionel propriétaire de la marque déposée « GenCode », le nom du langage de balisage générique pour les systèmes de composition. GenCode, LaTeX, BookMaster et VAX Document sont tous des exemples de « modèles d'état » (*state models*) : chaque changement d'état du système de composition est provoqué par un code formé d'une chaîne unique de caractères. Par exemple la commande **.bold** change l'état du formateur qui imprimera désormais les caractères en gras, jusqu'à ce qu'il rencontre une autre commande qui remettra le formateur dans l'état correspondant aux caractères normaux. Cependant, comme le montre le tableau 1, les codes utilisés sont toujours propres à un langage particulier.

Si l'on considère l'ensemble des structures possibles comme une arborescence, alors le langage de balisage décrit un document comme une réalisation d'un langage dont les règles de production sont décrites par une grammaire dite *context-free* dans la classification de N. Chomsky (c'est-à-dire que les règles de production ne dépendent pas du contexte d'utilisation). En définissant la grammaire, on peut établir la classe des structures possibles des documents. Ainsi, SGML n'est pas un « langage de balisage » dans le même sens que `troff` et TeX. SGML est un langage de spécification ou un méta-langage, qui permet la définition de règles pour la création d'un nombre infini de langages de balisage. SGML ne s'occupe pas de la composition des documents en question. Un principe fondamental dans la conception de SGML est de rendre le langage de balisage **indépendant** du logiciel de présentation ou du système de compostion. Des documents balisés en SGML peuvent être échangés entre des systèmes et installations différents.

Description	SGML (DTD AAP)	BookMaster	LaTeX	VAX Document
Commandes de sectionnement				
Niveau 0	pas disponible	:H0.	\part	<PART>
Niveau 1	pas disponible	:H1.	\chapter	<CHAPTER>
Niveau 2	<sec>	:H2.	\section	<HEAD1>
Niveau 3	<ss1>	:H3.	\subsection	<HEAD2>
Niveau 4	<ss2>	:H4.	\subsubsection	<HEAD3>
Niveau 5	<ss3>	:H5.	\paragraph	<HEAD4>
Niveau 6	<ss4>	:H6.	\subparagraph	<HEAD5>
Nouveau paragraphe	<P>	:P.	\par	<P>
Mise en valeur du texte et notes				
Mise en valeur type 0 (normal)	texte normal	:HP0.	texte normal ou {...}	texte normal
Mise en valeur type 1 (italique)	<e1>texte</e1>	:HP1.texte:EHP1.	{\it texte}	<EMPHASIS>(texte)
Mise en valeur type 2 (gras)	<e2>texte</e2>	:E2.texte:E2.	{\bf texte}	<EMPHASIS>(texte\BOLD)
Mise en valeur type 3 (gras-italique)	<e3 >texte</e3>	:HP3.texte:EHP3.	pas utilisé tel quel	pas utilisé tel quel
Citation	<Q>texte</Q>	:Q.texte:eQ.	``text''	<QUOTE>(texte)
Note en bas de page	<FN>texte</FN>	:FN.texte:eFN.	\footnote{texte}	<FOOTNOTE>(texte)
Les listes				
Liste numérotée	<l1>	:OL.	\begin{enumerate}	<LIST>(numbered)
Liste non-numérotée	<l2>	:UL.	\begin{itemize}	<LIST>(unnumbered)
Entrée dans la liste	<LI>	:LI.	\item	<LE>
Liste de description	<DL>	:DL.	\begin{description}	<DEFINITION_LIST>
Terme de la liste de description	<DT>	:DT.	\item[terme]	<DEFLIST_DEF>
Définition de la liste de description	<DD>	:DD.	texte	<DEFLIST_ITEM>
Les caractères mathématiques et spéciaux				
Grec (exemple α)	α	α	α	<MATH_CHAR>(ALPHA)
Symbole mathématique (exemple $\approx$)	≈	&app.	$\approx$	<MATH_CHAR>(approx)
Formule dans le texte	<f>formule</f>	:F.formule:EF.	$formule$ ou \(formule\)	<MATH>(formule)
Formule mise en valeur	<fd>formule</fd>	:DF.formule:EDF.	$$formule$$	<MATH>(display)formule
Exposants	<f>^{texte}</f>	:SUP.texte:ESUP.	$^{\mathrm{texte}}$	<SUPERSCRIPT>(texte)
Indices	<f><inf>texte</inf></f>	:SUB.texte:ESUB.	$_{\mathrm{texte}}$	<SUBSCRIPT>(texte)
Accents (exemple é)	é	&ea.	\'e	<MCS>(small_e_acute)
Blanc insécable		&rbl.	~	Pas utilisé tel quel

Table 1. Une comparaison de quelques systèmes de traitement de texte

Une conséquence de l'utilisation d'un balisage générique est qu'avec SGML il n'existe **pas** de balises pour spécifier la « mise en page », car celle-ci ne concerne pas la structure logique du document. Il n'y a pas de commandes pour indiquer que l'on veut une « nouvelle page », une « nouvelle ligne » ou tracer un « filet ». Ces éléments de mise en page sont plutôt des caractéristiques de certaines composantes ; ainsi, un titre du premier niveau peut, dans un style de présentation donné, démarrer une nouvelle page et un filet peut séparer le « titre » du « corps » du document.

1.1. Balisage SGML

Les balises SGML sont introduites dans un document en spécifiant pour chaque élément textuel sa position dans l'arborescence générale décrite formellement par la « Définition de Type de Document » *(DTD: Document Type Definition)*. Un document est une **réalisation** d'une classe de documents qui ont tous la même structure (*document instance*). Une fois qu'un élément a été repéré, le balisage se fait avec une **balise d'ouverture** (*start-tag*) et une **balise de fermeture** (*end-tag*). Sous sa forme la plus simple, une balise SGML consiste en un délimiteur de balise d'ouverture (`<`) suivi par **l'identificateur générique** (*generic identifier*), suivi par le délimiteur de balise de fermeture (`>`) ; exemple :

```
<Titlep>
```

Dans cet exemple `Titlep` est l'identificateur générique. Le nom d'un identificateur SGML ne dépend pas de la « casse » des caractères qui le composent, c'est-à-dire qu'il peut être saisi soit tout en majuscules `<TITLEP>`, soit tout en minuscules `<titlep>`, soit en un mélange des deux `<TiTLeP>` ; il s'agit toujours du même identificateur.

Une balise peut se trouver n'importe où sur la ligne, donc pas forcément au début. Un élément peut avoir un contenu vide, (par exemple la balise `<date>` utilisée toute seule spécifie que la date effective devra être fournie par le système de traitement de texte), ou il peut délimiter une partie du texte (par exemple `<P>` pour un paragraphe) qui a des données de type caractères (avec ou sans autres balises) comme contenu.

Une balise de fermeture a le même identificateur générique que la balise d'ouverture, mais est précédée par le délimiteur d'ouverture de balise de fermeture « `</` » ; exemple : `</TITLEP>`.

La figure 3 montre les composants des balises.

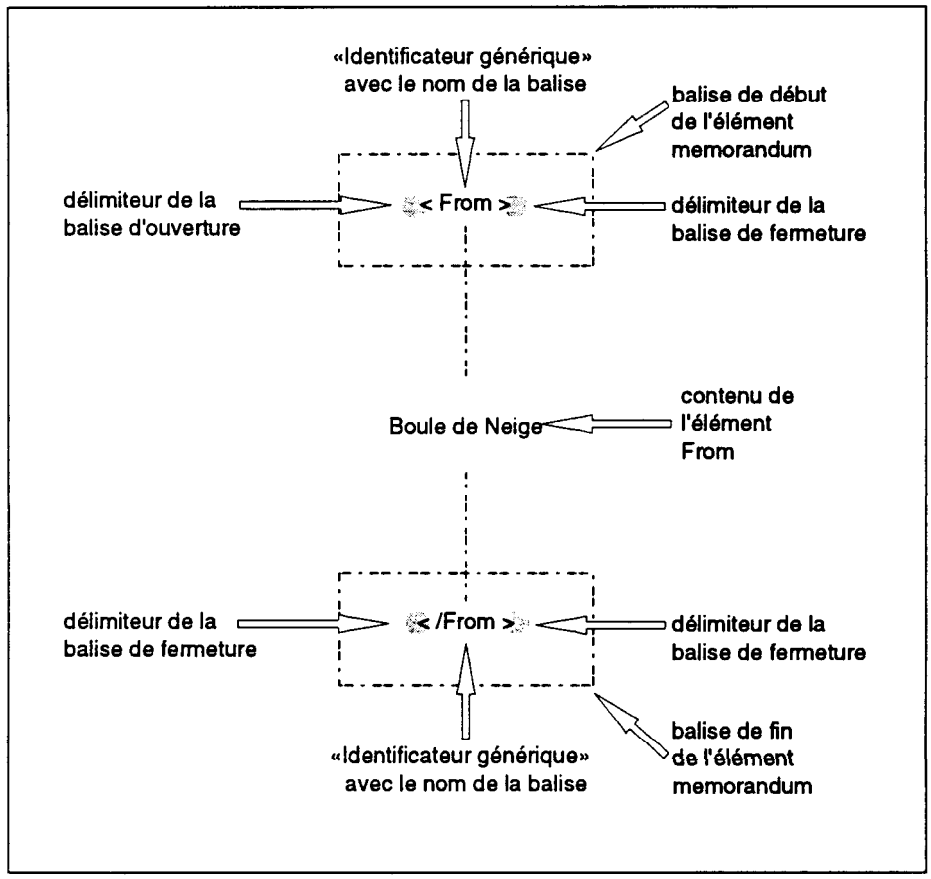

Figure 3. Les composants des balises

Dans sa forme la plus générale, une balise SGML consiste en un délimiteur d'ouverture, un identificateur générique, une **liste d'attributs**, un délimiteur de fermeture et du texte appartenant à la balise. Exemple :

```
<Titlep format=standard statut=public>Ce document ...
```

Il existe deux syntaxes possibles pour spécifier les attributs d'une balise. La forme la plus courante est constituée par l'identificateur de l'attribut, un « = », et une valeur ne contenant pas de caractères blancs. Mais, on peut aussi utiliser des valeurs d'attributs seules. Exemple :

```
<Titlep standard public>Ce document ...
```

Si la valeur d'un attribut contient autre chose que des caractères alphanumériques, elle doit être délimitée par des guillemets, comme :

```
<Titlep format="non standard" statut="secret" echelle="1.5">
```

Les choix des délimiteurs « < », « > », « = » ainsi que par exemple la longueur maximale des identificateurs génériques sont définis dans la **syntaxe concrète de référence** de SGML. Tout utilisateur peut définir une autre syntaxe en modifiant une composante spéciale d'un document SGML qui s'appelle la **déclaration SGML**. Celle-ci est souvent omise et les valeurs nécessaires sont alors fournies par défaut par le système.

Une autre notion très importante en SGML est celle d'**entité**. En effet, un document peut contenir des éléments qui ne peuvent être saisis directement au clavier, par exemple des caractères grecs ou des symboles mathématiques. On peut également vouloir inclure des illustrations ou des photos, ou un morceau de texte provenant d'un autre document. Alors SGML a recours a une entité représentée par une chaîne de caractères, associée à l'élément à insérer en question. Si l'on veut, par exemple, saisir 'GUT' et obtenir dans le texte final « Association GUTenberg », on pourra definir cette entité comme suit :

```
<!ENTITY GUT "Association GUTenberg">
```

Pour les illustrations et les documents externes, on procède de façon analogue. Dans ce cas, le champ de définition de l'entité contiendra l'indentificateur du fichier dans le système d'exploitation de la machine et indiquera le type des données qui s'y trouvent. Dans la figure 4, on trouve plusieurs types d'entités :

- définition d'une illustration PostScript (2^e et 3^e lignes)
- définition d'un texte externe (4^e et 5^e lignes)
- référence à des lettres accentuées (`&agrave;`, `&eacute;`, ...)

Comme le montre ce dernier cas, à l'intérieur d'un document une référence à une entité indique l'endroit ou l'entité doit être insérée, par exemple `&GUT;`. Notez la perluète (&) et le point-virgule (;) qui délimitent respectivement le début et la fin d'une référence à une entité. Il faut aussi savoir, qu'à l'inverse des noms de balises, les noms d'entités ne peuvent pas être spécifiés invariablement en majuscules ou en minuscules (les identificateurs `&GUT;`, `&gut;` et `&Gut;` font référence à trois entités différentes).

1.2. DTD : définition de type de document

La définition de type de document (DTD) contient l'ensemble **complet** de toutes les balises pour une classe donnée de documents. L'idée est que les utilisateurs d'une application (par exemple pour l'échange de textes entre les auteurs et une ou plusieurs maisons d'édition) analysent leurs besoins **communs** et que, de cette discussion, sorte un ensemble de types et de structures de documents à considérer. Les parties en présence se mettent alors d'accord sur les éléments logiques indispensables, comme les chapitres, les paragraphes, les notes de bas de page, etc. Notons que le choix des balises peut rester différent pour chaque application.

L'utilisation de la DTD offre l'avantage de ne pas normaliser les noms des balises et les structures. Ainsi tout le monde peut définir son propre langage de balisage tout en profitant des outils et produits SGML, dont le nombre croît chaque jour. Toutefois une analyse approfondie s'impose avant que des applications puissent échanger librement des documents.

Une DTD définit les objets suivants :

- Les *noms* de tous les éléments autorisés dans un document d'une classe donnée.
- Le *nombre d'occurrences* d'un élément.
- L'*ordre* dans lequel les éléments doivent apparaître.
- Si les balises d'ouverture ou de fermeture peuvent être *omises* ou pas.
- Les *conventions de saisie* afin d'alléger l'introduction des balises. Par exemple, une ligne blanche suivie d'une ligne décalée de 3 espaces blancs peut être interprétée comme une balise d'ouverture d'un paragraphe.
- Le *contenu* de tous les éléments, c'est-à-dire les noms de tous les autres identificateurs génériques qui peuvent être contenus dans l'élément, jusqu'au niveau des données de type caractère.

- Les *attributs* possibles pour chaque balise et leurs valeurs par défaut.
- Les noms de toutes les *entités* prédéfinies, auxquelles l'utilisateur a accès.

Une DTD ne contient pas d'information sur le traitement d'un document. Elle ne définit pas non plus quels délimiteurs doivent être utilisés pour le balisage. Un programme d'analyse *(parser)* vérifie la structure d'un document SGML et contrôle si son balisage est conforme aux noms et règles définis dans la DTD.

Lorsqu'on parle de SGML, il est très important de bien spécifier quelle DTD est utilisée. Quelques produits SGML n'offrent pas la possibilité d'utiliser une autre DTD que celle fournie avec le produit. Un exemple d'une DTD qui a été dévelopée pour des articles scientifiques est celle de l'Association des éditeurs américains (AAP) [5].

1.3. Exemple de document SGML

La figure 4 montre le début du présent article balisé en SGML. Les textes saisis sous forme SGML sont indépendants de tout logiciel ou configuration d'ordinateur, car ils ne contiennent pas de séquence de caractères de contrôle spécifiques et le fichier est en ASCII.

Cette indépendance par rapport à un système quelconque fait à la fois la force et la faiblesse de SGML. Il est très beau d'avoir son document en format SGML, mais avant de pouvoir l'utiliser (par exemple, l'imprimer) on a besoin d'un programme de traduction qui transforme les balises SGML en commandes d'un système de traitement de texte.

1.4. SGML et hypertextes

Un des atouts de SGML est de permettre des applications multiples à partir d'un même texte. Il est facile d'ajouter une dimension supplémentaire à l'information présente dans un document SGML, dont le seul but ne doit plus se limiter à l'obtention d'une version imprimée sur papier. Considérons la notion d'hypertexte, qui est très à la mode de nos jours. Le fait qu'un document soit déjà saisi en SGML permet une extraction aisée de l'information pour la mettre sous forme d'hypertexte.

Un projet intéressant où SGML est appliqué à un hypertexte est « Hytime » : il est actuellement en train d'être adopté comme norme

```
<!DOCTYPE article SYSTEM "AAPARTCL DTD  *" [
<!ENTITY FIGSPDL  SYSTEM "SPDL     EPS  *" CDATA EPS>
<!ENTITY FIGDSSSL SYSTEM "DSSSL    EPS  *" CDATA EPS>
<!ENTITY EXTPOST1 SYSTEM "POSTSCR1 SGML *">
<!ENTITY EXTPOST2 SYSTEM "POSTSCR2 SGML *">
]>
<article>
<atl>Introduction &agrave; SGML, DSSSL et SPDL
<DATE><yr>22 novembre 1991
<AU>Michel Goossens et Eric van Herwijnen
<Aff>CERN,
<odv>Division AS
<cty>Gen&egrave;ve 23
<pc>1211-CH
<cny>Suisse
<pubfm>
<cdn>AS/91-3
</pubfm>
<ABS>
<p>Cet article passe en revue quelques aspects de la norme
ISO 8879 SGML, le langage normalis&eacute; de balisage
g&eacute;n&eacute;ralis&eacute;, et la situe par rapport
&agrave; d'autres projets de normes pour la description des
documents sur ordinateur : DSSSL pour la mise en
page et SPDL pour la visualisation.
</ABS>
<BDY>
<sec id="h1sgml"><st>Inter&ecirc;t de la norme SGML</st>
<P>
SGML &mdash; langage normalis&eacute; de balisage
g&eacute;n&eacute;ralis&eacute;, ou <e3>Standard Generalised Markup
Language</e3> en anglais &mdash est une norme internationale ISO
(organisation internationale de normalisation dont le
si&egrave;ge est &agrave; Gen&egrave;ve), qui d&eacute;finit un
formalisme de sp&eacute;cification pour le <e2>balisage</e2> de
documents sous forme &eacute;lectronique afin de faciliter
```

Figure 4. Exemple d'un document balisé en SGML (DTD AAP-Article)

américaine et internationale (ISO DIS 10744) pour la réprésentation structurée de l'information hypermedia.

L'idée est qu'un document contient un ensemble d'événements dépendant du temps, par exemple de l'audio, de la vidéo, des images figées, de la danse ... Les documents et les événements sont connectés par un ensemble d'« hyper-liens » (*hyperlinks*). Afin de permettre une description normalisée de ces liens, HyTime utilise un système d'adressage puissant.

Pour décrire la synchronisation des événements, on a utilisé le modèle temporel propre à la musique. Dans la musique écrite, le temps est un élément « virtuel », dans le sens qu'on n'utilise que des notes, des variations de tempo rélatives, etc. Le temps « réel » d'une œuvre musicale dépend du musicien qui la joue. Ce modèle a été développé pour le « Langage normalisé de description de la musique » (SDML, ISO DP 10743), qui est une application HyTime. Dans HyTime, la durée de chaque événement est mesurée en unités de temps virtuel *(vtu's)*. Celles-ci sont reliées au temps réel par une « baguette » qui exprime le temps virtuel comme une fonction générale du temps réel. Des indications de tempo qui ne sont pas précises, comme lent ou rapide, sont également possibles.

Ce modèle peut être décrit en utilisant seulement la fonctionalité la plus simple de SGML, comme les éléments, les entités et les attributs. Des liens hypertextes, connectant deux points dans un même document ou dans des documents différents, sont réalisés à l'aide d'éléments spéciaux appelés *link*. Un exemple de séquence Hytime multi-media balisée en SGML est donnée dans la figure 5 :

```
<mmseq id=dgiovanni baton=haitink>
<ces><ce><musicdur><vtu>1<ce><musicdur><vtu>1<musicdur><vtu>1<ce></ces>
</mmseq>
<baton id=haitink>
<tempo><musicdur><vtu>1<realdur>5400<tempo><musicdur><vtu>2<realdur>2800
</baton>
```

où :

mmseq	=	séquence multi-média
ces	=	séquence d'événements centraux
ce	=	événement central
musicdur	=	durée d'un événement en unités virtuelles (vtu's)
realdur	=	durée d'un événement en temps réel

Figure 5. Example de balisage d'une séquence Hytime

HyTime peut décrire n'importe quelle situation dépendant du temps, comme par exemple des simulateurs de conduite ou de vol, le format CD-ROM et CD-I. Le travail sur Hytime est coordonné par Charles Goldfarb [10] chez IBM (Almaden) et Steve Newcomb [11] à l'Université de Florida State.

Un exemple de système hypertexte disponible dans le commerce et capable d'utiliser directement des documents SGML est « Dynatext »[9] de Electronic Book Technologies.

Dans le domaine de la recherche nous pouvons mentionner A. Fountain et ses collaborateurs [12] à l'Université de Southampton qui proposent d'utiliser SGML pour décrire de façon dynamique les liens crées par un lecteur pendant qu'il parcourt un document hypertexte.

2. DSSSL– format de présentation de document et langage de spécification

SGML décrit la structure logique d'un document, mais laisse à l'utilisateur le soin de développer ses propres outils pour visualiser cette structure sur une page imprimée ou à l'écran. La séparation du contenu et de la forme permet des utilisations multiples des mêmes données. Plusieurs programmes de traduction (ad hoc) existent pour convertir les balises SGML en codes pour des systèmes de traitement de texte. Cependant, il est parfois nécessaire d'obtenir une correspondance point par point au niveau de la sortie de deux documents échangés. Jusqu'à ce jour il n'existait **pas** de méthode standard pour décrire la présentation externe des éléments sur un support de sortie. DSSSL (*Document Style Semantics and Specification Language*) tente d'apporter une réponse à ce problème.

Un des objectifs de DSSSL [13] est « de fournir un moyen formel et rigoureux d'exprimer la gamme des spécifications pour la production de documents, y compris la typographie de haute qualité, requise par l'industrie des arts graphiques ». DSSSL part d'une « application SGML et propose un formalisme normalisé pour les spécifications de présentation. Beaucoup d'utilisateurs ont besoin d'une approche unifiée pour échanger l'information de présentation et autres traitements ». On pourrait même envisager d'étendre DSSSL à l'extraction de l'information contenue dans des documents SGML pour la charger dans une base de données. Autrement dit, DSSSL propose une norme pour décrire la présentation externe des documents.

DSSSL associe des spécifications de présentation aux éléments logiques d'un document SGML comme le montre la figure 6.

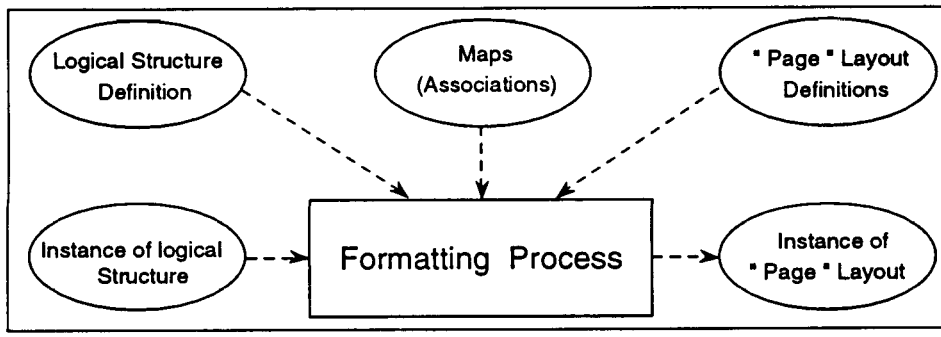

Figure 6. Association de l'information de présentation aux éléments logiques (DSSSL)

Cette association se divise en deux étapes :

1. Une transformation de type général pour changer la structure du document (en ajoutant par exemple une table des matières, en créant une bibliographie, en satisfaisant des références internes et externes). Le résultat de cette transformation est un document intermédiaire qui peut exister ou non sous forme réelle.
2. Une transformation spécifique basée sur la sémantique. La zone de sortie est définie à l'aide du langage de spécification, ce qui permet une sortie ultérieure sur papier ou un autre support.

Un document DSSSL contient:

1. Le texte saisi en SGML, balisé d'après les règles de la DTD du document original.
2. La DTD qui correspond à la structure du document virtuel intermédiaire.
3. La spécification DSSSL, qui associe à chaque élément logique une valeur sémantique d'après l'architecture du document DSSSL.

Comme c'est le cas avec SGML, on a besoin d'un logiciel pour traiter l'information présente dans les documents DSSSL. Un formateur DSSSL sera capable de générer automatiquement la spécification DSSSL et pourra

l'appliquer aux documents qui se conforment à SGML. La sortie générée par l'étape « composition » peut être un fichier d'entrée pour un logiciel de traitement de texte ou bien une sortie en PostScript ou « SPDL », le langage standard de description de page (voir ci-dessous). Depuis février 1991, DSSSL est au stade de projet de norme internationale.

3. SPDL - Langage standard de description de page

SPDL [14] (*Standard Page Description Language* ou langage normalisée de description de page) définit « un langage pour la spécification d'un format de visualisation (par exemple imprimé ou présenté à l'écran) des documents électroniques, contenant du texte en noir et blanc, à plusieurs niveaux de gris, ou entièrement en couleur, des images, des formes géométriques ». C'est typiquement le format pour un document dans sa phase finale. SPDL permet une reproduction efficace de la « représentation » calculée dans la phase composition et mise en page.

SPDL s'inspire fortement de PostScript et de son prédécesseur Interpress. En effet, Adobe et Xerox ont dominé l'effort de normalisation qui a engendré SPDL. Ce projet de norme ISO définit, indépendamment du support de sortie, un formalisme pour décrire la présentation de documents contenant du matériel textuel et graphique.

Un document SPDL a une *structure* et un *contenu*. La structure d'un document SPDL est hiérarchique, le niveau le plus haut étant le **document**. Un document peut contenir des **ensembles de pages** (*pagesets*) et des **pages**. Une page peut contenir des **images** (*pictures*) et des **séquences de tokens (jetons)** (*tokensequences*). Une description de cette structure sous forme de DTD SGML est proposée dans le texte du projet de norme SPDL et sous forme schématique dans la figure 7. Quant au contenu d'un document SPDL, il consiste en données contenues dans des structures de suites de tokens. L'interprétation des tokens se fait à l'aide d'une machine à états virtuels. La réalisation des images est fonction d'un système de coordonnées, d'encres et de régions de « découpe » (*clipping*) et utilise trois catégories d'opérations graphiques : les chaînes de caractères textuels, les images bitmap et les formes géométriques.

La figure 8 montre le traitement d'un document SPDL.

Le présentation à la sortie est décidée (calculée) dans la phase de composition et de mise en page (*Layout and Composition Process*), qui

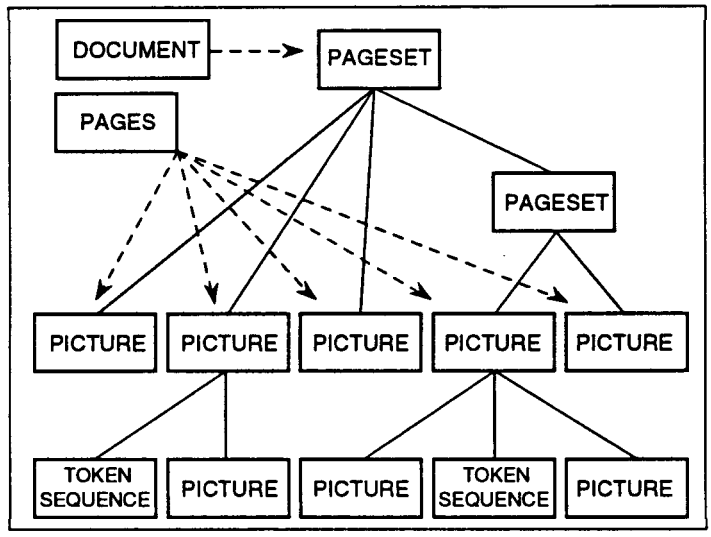

Figure 7. La structure d'un document SPDL

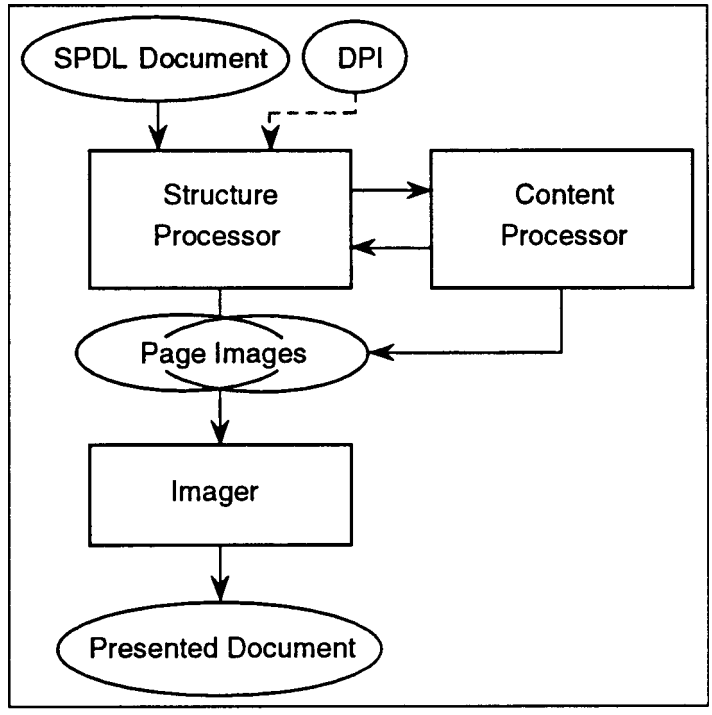

Figure 8. Le modèle de traitement d'un document SPDL

peut être un formateur DSSSL. Ces décisions comprennent la sélection des polices des caractères, le placement des caractères dans le plan de sortie et la présentation des graphismes.

Le logiciel de visualisation prend le document en format SPDL et essaie de garantir la meilleure approximation possible en tenant compte des contraintes du support de présentation, des calculs réalisés pendant la phase composition et mise en page.

Comme DSSSL, SPDL en est encore au stade de projet de norme internationale. Cependant, grâce à sa ressemblance aux langages de description de page existants, on s'attend à voir des implementations pratiques en SPDL bien avant celles en DSSSL.

4. Conclusion

Ensembles, SGML, DSSSL et SPDL proposeront les outils suivants :

- une norme (SGML) pour spécifier la structure logique d'une classe de documents (DTD), ainsi qu'un moyen de représenter un document comme réalisation de cette classe ;
- une norme (DSSSL) pour spécifier la présentation visuelle associée à un document appartenant à une classe donnée de documents SGML ;
- un norme (SPDL) pour réaliser une visualisation concrète d'un document spécifique.

La figure 9 montre comment SGML, DSSSL, SPDL et FONTS seront utilisés ensemble (voir [15] et [16]).

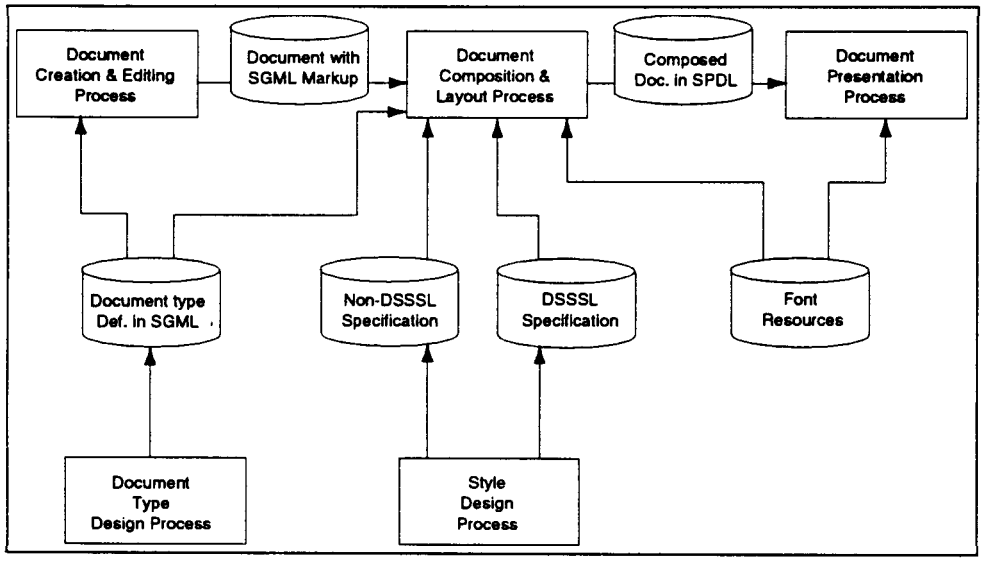

Figure 9. Modèle operationel des normes ISO pour le traitement de texte

Ainsi que nous l'avons évoqué plus haut, DSSSL n'est pas encore une norme internationale. De plus, du fait de la complexité de son langage de spécification, il est difficile de le coder à la main. DSSSL ne sera utilisable que lorsque des formateurs adéquats seront disponibles, ce qui peut prendre encore quelques années. Cela ne signifie par pour autant que l'on ne puisse commencer à utiliser SGML immédiatement. En effet, même si la façon dont les documents SGML sont utilisés n'est pas, pour l'heure, normalisée, tous les produits SGML actuellement disponibles viennent avec leur propre système de traduction, qui transforme les balises SGML en commandes d'un système de traitement de texte. Les utilisateurs de SGML peuvent donc déjà aujourd'hui bénéficier de la plus-value qu'offre SGML dans les domaines de la portabilité, la généralité, l'indépendance de la forme et de sa panoplie de possibilités d'utilisation.

Remerciements

Nous tenons à remercier Anders Berglund (ISO, Genève), Michèle Jouhet(CERN) et Laurent Le Gal (CERN) pour leurs commentaires et suggestions et Arlette Couder (CERN) et Claude Rigoni (CERN) pour leur aide technique dans la réalisation des figures.

Références bibliographiques

[1] Organisation internationale de normalisation. *Langage normalisé de balise généralisé (SGML).* ISO 8879-1986(F), ISO Genève, 1986.

[2] E. van Herwijnen. *Practical SGML.* Wolters-Kluwer Academic Publishers, Boston, 1990.

[3] D. Vignaud. *L'édition structurée des documents.* Éditions du Cercle de la Librairie, Paris, 1990.

[4] Norsk Data. *Nortext-100 Typographic Function Codes.* ND-61.029.1 EN, 1986.

[5] American National Standards Institute. *American National Standard for Electronic Manuscript Preparation and Markup.* ANSI/NISO Z39.59-1988, 1988.

[6] IBM Corporation, Boulder. *Document Composition Facility BookMaster - Version 3.* SC34-5009-03, 1990.

[7] L. Lamport. LaTeX, *A document preparation system.* Addison Wesley, Reading, 1986.

[8] Digital Equipment Co. *VAX Document, Using Global Tags.* AA-JT84C-TE, 1991

[9] Electronic Book Technology, Inc. *Dynatext, Electronic Book Indexer/Browser.* Providence, Rhode Island, 1991.

[10] C.F. Goldfarb. HyTime: A standard for structured hypermedia interchange. *IEEE Computer*, pages 81–84, august 1991.

[11] S.R. Newcomb. *SIGhyper, SGML User's Group SIG on hypertext.* TechnoTeacher, Tallahassee, FL, July 1991

[12] A.M. Fountain, W. Hall, I. Heath, and H.C. Davis. *Microcosm: an open Model for hypermedia with dynamic linking.* Cambridge University Press (proceedings of ECHT '90), 1990.

[13] International Organization for Standardization, S. Adler (Editor). *Document Style Semantics and Specification Language.* ISO DP 10179, ISO Geneva, 1991.

[14] International Organization for Standardization, S. Strasen and M. Fowley (Editors). *Standard Page Description Language.* ISO DP 10180, ISO Geneva, 1991.

[15] International Organization for Standardization. *Font information interchange* (trois parties). ISO 9541-1,2,3, ISO Geneva, 1991.

[16] D. Dardailler, « Normes et fontes », *Cahiers GUTenberg*, n° 4, décembre 1991, 2–9.