

Cahiers **GUT** *enberg*

☞ INTÉGRATION DE GRAPHIQUES DANS DU TEXTE

☞ Sarra BEN LAGHA, Mohamed BEN AHMED

Cahiers GUTenberg, n° 21 (1995), p. 53-67.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1995__21_53_0>

© Association GUTenberg, 1995, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

Intégration de graphiques dans du texte

Sarra BEN LAGHA et Mohamed BEN AHMED

Laboratoire RIADI École Nationale des Sciences de l'Informatique

16, rue 8010 Montplaisir 1002 Tunis (TUNISIE)

Tel. +216 (1) 781.419 Fax. +216 (1) 785.659

e-mail : *bellagha@ensi.rnrt.tn*

Résumé. L'édition des documents techniques est une opération complexe mettant en oeuvre plusieurs compétences et divers outils. Pour que l'échange de documents dans un environnement hétérogène soit possible, il est nécessaire que ces documents soient normalisés. S'agissant de documents composites, chacun des composants (textuel, graphique,...) doit respecter un ou plusieurs standards, et l'intégration de ces différentes composantes doit être, elle aussi, normalisée. On parle alors dans ce cas de documents multi-normes, multi-plateformes. Cet article passe en revue les notions de base de la norme de structuration de documents SGML, et de la norme graphique CGM, avant de présenter deux méthodes pour l'intégration de graphiques dans les documents structurés normalisés SGML.

Mots-clés: Document, Graphique, Intégration, Norme, SDIF, SGML, DTD, CGM, GKS.

1. Les documents techniques

On distingue généralement les documents techniques des documents tout court. En effet, il existe deux grandes familles de documents: les documents administratifs et les documents techniques. Bien que les solutions adaptées à la gestion de ces différents documents soient proches les unes des autres, elles diffèrent sur quelques points. Une grande différence apparaît au niveau de la structure même de chacune de ces classes. Les documents administratifs (lettres, rapports,...) sont conçus pour présenter un certain nombre d'informations, de conclusions et de décisions dans un format et un style plus ou moins variables en fonction de leurs auteurs. Les documents techniques sont par contre moins personnalisés, plus structurés et plus complexes. Destinés à exposer et à expliquer certains concepts (manuels d'utilisation, guides techniques,...) ou à servir de référence dans des situations de crises (plans, manuels de maintenance,...), leur production peut être une procédure complexe mettant en jeu beaucoup de personnes, de professions et différentes méthodes. Ce sont des documents composites faisant intervenir un certain nombre d'outils informatiques de catégories diverses: éditeurs de texte, éditeurs graphiques, tableurs,... Une seconde différence se situe au niveau des modifications des documents originaux après stockage. Les documents administratifs évoluent peu. Ils sont parfois

annotés ou utilisés comme point de départ pour la conception de nouveaux documents. Les plans et les documents techniques évoluent par contre au même rythme que les produits qu'ils accompagnent. Dans ce domaine, la notion de révision associée à une nouvelle version d'un produit est primordiale. On parle en effet de documents révisables ou retraitables. La réutilisation d'un document consiste à inclure les informations qu'il contient dans d'autres documents. C'est le cas d'une information technique qui figure, par exemple, aussi bien dans un document de conception, que dans le manuel d'utilisation ou de formation pour le même produit. Or, pour réutiliser ces informations dans d'autres documents, il doit être possible d'introduire les documents dans d'autres systèmes de traitement de texte et d'autres outils de CAO, de PAO, etc. D'où le besoin de disposer d'un format commun d'échange de documents. Un tel format ne peut cependant être obtenu que si des normes d'échange de documents existent et que les utilisateurs acceptent de s'y conformer.

2. La norme d'échange SDIF

La norme internationale ISO 9069, spécifie une structure de données appelée format SDIF (Sgml Document Interchange Format). SDIF permet de condenser dans un flux de données, en vue d'un échange, un document conforme à la norme SGML (Standard Generalized Markup Language) ISO 8879, éventuellement stocké sous forme de plusieurs entités, d'une façon telle que le destinataire puisse reconstituer les différentes entités [1]. SDIF permet par ailleurs d'inclure des documents connexes dans le flux de données, tels que lettres explicatives, formulaires de transmission, fiches de catalogue, procédures de formatage ou le profil fonctionnel exigé par une architecture de documents. D'après les spécifications de la norme ISO 9069:1988 *traitement de l'information – facilités de support SGML – Format d'échange de documents SGML (SDIF)*, ce format ne doit être utilisé que pour l'échange de documents SGML, tels que définis par la norme ISO 8879 [2], entre systèmes SGML. L'échange peut se faire par voie de communications de données dans l'environnement d'interconnexion de systèmes ouverts (OSI) ou dans d'autres types d'environnements, ou encore par échange des supports de stockage. Cependant, bien que destiné à assurer l'échange de documents SGML entre systèmes hétérogènes, le format SDIF peut être utilisé pour la transmission de documents bureautiques conformes à la norme ODA (Office Document Architecture), à condition que ces derniers respectent la représentation selon la norme SGML des documents ODA, tels que définis dans la norme internationale ODA [3]. Ainsi, tout document normalisé SGML ou ODA peut être échangé entre différents systèmes conformément à la norme SDIF. Dans ce qui suit nous ne nous intéresserons pas à la norme ODA.

3. La norme SGML

La norme internationale ISO 8879:1986 décrit les spécifications d'un langage de description de document nommé SGML, Standard Generalized Markup Language, soit *langage normalisé de balisage généralisé*. SGML peut être utilisé dans l'édition qui s'étend de l'édition traditionnelle 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 cherche à la fois la possibilité de relecture par les humains et de transfert vers des systèmes de publication [1]. SGML définit des méthodes pour le balisage de documents permettant de préserver l'information essentielle quand les documents sont échangés. Un schéma de balisage défini en SGML, décrit la structure logique d'un document. Cette information peut ensuite être encapsulée, transférée, et reconstituée en même temps que le contenu du document [4]. La norme SGML n'est pas en elle-même un schéma de balisage, mais une méthodologie pour créer des schémas syntaxiques de balisage généralisé, pour différents types de documents et applications. SGML normalise les méthodes utilisées dans la préparation des documents. Elle fournit un fondement sur lequel les applications peuvent développer leurs propres schémas de balisage syntaxiques. En utilisant ces méthodes, tout modèle de balise d'une application peut avoir sa propre syntaxe et rester compatible avec les autres applications. SGML fournit des méthodes pour interpréter tout schéma de balisage spécifique appliqué, de telle sorte que l'échange soit possible même quand on traverse les frontières des applications [4]. Tout document SGML n'est qu'une instance d'une classe de documents. Une classe est un modèle définissant la structure logique (chapitres, paragraphes, sections,...) par un schéma de balisage normalisé appelé DTD: Document Type Definition. Ainsi, la création d'un document SGML consiste à choisir ou à créer une définition de type de documents (DTD) qui définit la syntaxe et les balises de la classe de documents ayant la même structure logique que le document en question, puis à baliser ce dernier (en utilisant n'importe quel système de traitement de texte) selon les critères définis par la DTD. La tâche la plus difficile dans l'utilisation de SGML réside dans la conception des DTDs. Ces dernières doivent en effet refléter exactement la structure des documents, qu'ils soient déjà existants dans le but de les normaliser, ou futurs lorsqu'on décide d'adopter la norme. Ce processus consiste en fait à analyser le document de façon à :

- identifier ses composants tels que: en-têtes, figures, tableaux et notes de bas de pages...
- créer la table des matières, les listes des figures et des tableaux,
- créer les index.

Un exemple détaillé d'analyse de documents et de création de DTDs peut être consulté dans [5].

4. Le traitement des graphiques dans SGML

Les images et les graphiques sont des composants logiques particuliers d'un document. Dans une DTD SGML ils sont généralement identifiés comme éléments flottants. Créés par des outils graphiques dédiés et ayant des formats non textuels, ils sont généralement collés dans un document à des positions bien définies. Une figure est insérée dans un document technique pour illustrer un certain nombre de concepts. Elle possède généralement un corps, un titre et une légende ou description. Ainsi, elle peut être représentée dans une instance de document SGML par les balises suivantes:

```
<fig>  
<corpsfig>  
<titfig> Titre de la figure  
<legfig> Légende pour la description de la figure  
</fig>
```

Dans la DTD, le corps de la figure sera défini comme une entité réservée sans contenu (EMPTY) puisque le corps de la figure sera contenu dans un fichier graphique et collé simplement à la position indiquée par le balisage dans le document. SGML offre outre la syntaxe permettant de définir des schémas de balisage, la notion d'attributs. Pour insérer une figure dans un document textuel, il est en principe utile de connaître un certain nombre de caractéristiques qui permettraient de lui prévoir la place nécessaire dans le document d'une part, et de la positionner par rapport aux autres éléments qui l'entourent, d'autre part. Ainsi, on pourrait utiliser les attributs: **id**, **delim**, **position**, **largeur**, **aligner**, **ligne**. Pour l'élément réservé, deux attributs peuvent être utiles pour définir la largeur (**larg**) et la hauteur (**haut**) de la figure à insérer, cette dernière étant évidemment contenue dans un fichier externe auquel on fait référence au début de la DTD. Une figure peut être référencée tout au long du document, la valeur de l'attribut identificateur (id=) sert alors à cette fin et doit être unique dans le document. L'attribut de délimitation (delim=) peut prendre une des trois valeurs: "cadre", "bordure" ou "absent". La valeur "cadre" indique que la figure doit être contenue dans un cadre. La valeur "bordure" indique qu'une ligne sera tracée avant et après la figure. La valeur "absent" indique que la figure ne sera ni encadrée ni comprise entre deux lignes. La valeur par défaut lorsque cet attribut n'est pas explicitement utilisé est la valeur "absent". L'attribut (position=) prend une des trois valeurs: "haut", "fixe" et "bas". La valeur "haut" signifie que la figure sera collée en haut de la page suivante, "bas" signifie qu'elle sera collée en bas de la page courante s'il y a suffisamment de place pour le faire ou en bas de la page suivante dans le cas contraire. La valeur "fixe" indique que si sur la colonne courante il n'y a pas suffisamment de place pour coller la figure, le reste de la colonne (ou de la page) devra rester vide, et la figure devra être collée en haut de la colonne suivante. L'attribut (largeur=) prend la valeur "colonne" si la figure a la largeur d'une colonne, et la valeur "page" si la largeur de la figure est fonction

de celle de la page. L'attribut (`aligner=`) spécifie pour sa part comment la figure doit-elle être alignée horizontalement sur la page ou la colonne. Il prend une des valeurs "gauche", "centré" ou "droite", la valeur par défaut étant: "centré". Enfin, l'attribut (`ligne=`) indique si les éléments textuels éventuellement contenus dans le corps de la figure doivent être laissés tels quels (sans formatage) dans le cas où la valeur affectée à cet attribut est: "lignes", ou qu'ils doivent suivre le formatage de la portion textuelle du document dans laquelle la figure est insérée (la valeur est alors "continu"). Ainsi la partie balisée du document cité ci-dessus peut être précisée comme suit :

```
<fig id=FIG1 cadre >  
<corpsfig larg=100 haut=100 >  
<titfig> Titre de la figure  
<legfig> Légende pour la description de la figure  
</fig>
```

Ainsi, grâce à ces balises et ces attributs, la norme SGML permet d'inclure dans un document quelconque tout type d'image ou de figure contenu dans un fichier approprié et offre les moyens de préciser la façon avec laquelle l'élément graphique en question devra apparaître dans le document. Cependant, lors d'un échange, pour que le document soit correctement restitué à la destination, il faut transférer avec succès le fichier SGML d'une part, et le fichier graphique d'autre part. En appliquant le même raisonnement qui nous a mené à l'adoption de la norme SGML, nous concluons facilement qu'il faut recourir à une norme graphique. Donc, si nous essayons de résumer un peu ce que nous venons d'exposer jusque là, nous dirons que pour transférer un document composite entre deux systèmes SGML, nous devons commencer par écrire une DTD dans laquelle on spécifie la structure logique du document qui peut contenir des illustrations. Puis, utiliser les balises ainsi définies pour construire le document SGML conformément à la DTD en précisant pour chaque figure sa position, sa taille et ses attributs sans oublier d'établir le lien avec les différents fichiers externes qui contiennent réellement les illustrations selon un format normalisé. Et finalement transmettre le document selon le format SDIF et les illustrations selon la norme graphique utilisée après avoir vérifié que celle-ci est bien supportée par le système destinataire. Ceci représente en réalité le premier scénario d'utilisation des graphiques dans un document SGML. Un deuxième scénario peut cependant être envisagé. Il consiste à intégrer les graphiques dans l'instance de document en utilisant la norme graphique CGM.

5. La norme CGM

Le métafichier de données infographiques CGM offre un format de fichier approprié au stockage et à la réutilisation de données infographiques [6]. L'objectif de ce métafichier est de permettre la description, le stockage et la communication

d'informations graphiques de manière indépendante des appareils. Pour atteindre cet objectif, la norme CGM, définit la forme (syntaxe) et le comportement fonctionnel (sémantique) d'un ensemble d'éléments qui peuvent apparaître dans le métafichier CGM. La norme CGM définit plusieurs versions du métafichier. Une version étant définie par une grammaire formelle et des spécifications complémentaires. Les versions actuellement définies sont: la version 1, la version 2 et la version 3. Les éléments du métafichier CGM se répartissent en classes comme suit :

- les éléments délimiteurs qui séparent les structures significatives du métafichier,
- les éléments descripteurs de métafichier qui décrivent le contenu fonctionnel, les conditions par défaut, l'identificateur et les caractéristiques du métafichier CGM,
- les éléments descripteurs d'images qui établissent les modes d'interprétation des attributs pour chaque image,
- les éléments de contrôle qui permettent de modifier les limites d'images et la représentation des coordonnées,
- les éléments de primitives graphiques qui décrivent les composantes visuelles d'une image dans le métafichier CGM,
- les éléments attributs qui décrivent l'aspect des primitives graphiques et enfin,
- les éléments de segments: permettent de grouper des éléments et de les manipuler comme une seule entité.

Un métafichier de données infographiques est une collection d'éléments de cet ensemble. Les éléments délimiteurs BEGIN METAFIELD et END METAFIELD apparaissent une seule fois dans un métafichier complet qui peut cependant contenir un nombre quelconque d'éléments des autres classes, en fonction des besoins. Un métafichier doit être interprété afin d'afficher son contenu sur un appareil graphique, les éléments descripteurs fournissent alors à l'interpréteur les données nécessaires pour interpréter les éléments du métafichier et pour prendre les décisions correctes concernant les ressources requises pour la visualisation.

6. Spécification formelle et codage des métafichiers CGM

Les fonctionnalités offertes par le métafichier n'ont pas de lien avec la spécification d'un format particulier de codage. La norme CGM autorise à la fois des codages normalisés et des codages privés des éléments. Trois codages normalisés de la syntaxe du métafichier sont offerts, outre la spécification formelle, pour

répondre à des besoins différents: un minimum d'encombrement du métafichier, un minimum de traitement au niveau de la génération et de l'interprétation ou une souplesse maximale pour le lecteur humain ou l'éditeur du métafichier. Les transpositions entre codages normalisés sont, tout de même, possibles sans perte d'informations graphiques bien qu'une transposition ultérieure pour revenir au codage initial puisse dans certains cas ne pas donner exactement le même flux de données. Ces trois codages sont: le codage caractère, le codage binaire et le codage texte en clair. Chacun de ces trois codages est défini par une partie de la norme. Le codage caractère prescrit dans l'ISO/CEI 8632-2 est recommandé pour le transfert à travers des réseaux ne supportant pas les transferts binaires, et dans le cas où l'on a besoin de minimiser la taille du métafichier. L'ISO/CEI 8632-3 prescrit, quant à lui, un codage binaire qui exige moins de traitement de génération et d'interprétation sur de nombreux systèmes. En plus, il utilise des formats de données beaucoup plus proche des représentations internes de données des ordinateurs que les autres codages. Enfin, le codage texte en clair prescrit dans l'ISO/CEI 8632-4 est un codage qui peut être créé, lu et édité par des éditeurs de texte courants. Il est aussi recommandé pour les transferts à travers les réseaux qui ne supportent que le transfert de fichiers texte. La norme ISO/CEI 8632 définit alors deux formes de conformité de métafichiers infographiques. On parle de conformité fonctionnelle lorsque le contenu du métafichier correspond exactement à la spécification fonctionnelle définie dans la première partie de de l'ISO/CEI 8632 (selon des critères de conformité spécifiés par la norme elle même) et on parle de conformité complète lorsque le métafichier est fonctionnellement conforme à l'une des versions de la norme et est, de plus, conforme à l'un des types de codages spécifiés par l'ISO/CEI 8632-2, l'ISO/CEI 8632-3 ou l'ISO/CEI 8632-4.

7. CGM et SGML

Pour que des images CGM puissent être intégrées dans des documents SGML, elles doivent tout d'abord vérifier la conformité complète.

L'annexe E.2 de la norme SGML (qui ne fait cependant pas partie intégrante de la norme elle même), indique que l'intégration de telles images peut avoir lieu. Elle propose pour cela, la définition d'un élément graphique "graphic" avec un certain nombre d'attributs permettant de préciser le type du codage utilisé, le numéro de la figure concernée dans le fichier CGM (celui-ci pouvant en contenir plusieurs), la zone de la figure qu'on veut réellement utiliser, les facteurs d'échelles horizontale et verticale, l'orientation de la figure, etc.

Pour illustrer ceci, nous utiliserons le métafichier CGM de la figure 1 codé en mode "texte en clair", représentant une bicyclette [5].

Cette illustration CGM peut être directement fusionnée dans l'instance de document comme dans les deux exemples ci-dessous. Il est à signaler à ce niveau

que les valeurs réelles des attributs numériques dépend de l'implémentation (il peut s'agir de mm, de cm, de pouces, etc).

```
<fig id=FIG2 cadre >
<graphic sizex="100" x0="0.0" y0="0.2" x1="1.0" y1="0.65">
BEGMF "Bicycle";
...
ENDMF;
</graphic>
<titfig> Exemple d'illustration CGM directement
fusionnée dans le texte.
<legfig> Représentation de la vue définie par les
coordonnées (0.0,0.2) et (1.0,0.65), d'une fenêtre de
coordonnées (0.0,0.0) et (1.0,1.0), avec un facteur
d'échelle 100 (pour la largeur et la hauteur).
</fig>
```

```

BEGMF "Bicycle";
MFVERSION 1;
MFELEMLIST "DRAWINGPLUS";
REALPREC -0.3276700E+05,0.3276700E+05,6;
VDCTYPE REAL;
BEGPIC "PIC0001";
VDCEXT <0,0,0.0>,<0.1E+01,0.1E+01>;
BEGPICBODY;
LINE <0.6000005E-01,0.2350001>,<0.9900006,0.2350001>;
LINE <0.2150003,0.3550002>,<0.2950001,0.5350002>,<0.2950001,0.5650002>;
LINE <0.2950001,0.5450002>,<0.5950003,0.5450003>,<0.5950003,0.5450003>;
POLYGON <0.5450004,0.5650004>,<0.5950003,0.5450003>,
<0.6150002,0.5450003>,<0.6150002,0.5650004>,<0.5350003,0.5650004>;
LINE <0.2950001,0.5650002>,<0.2850001,0.5650002>,<0.2750002,0.5750002>,
<0.2750002,0.5850003>,<0.2850001,0.5950004>,<0.2850001,0.5950004>,
<0.3050002,0.5950004>,<0.3050002,0.5950004>,<0.3150002,0.5950004>,
<0.3150002,0.5950004>;
CIRCLE 0.215,0.355,0.12;
CIRCLE 0.655,0.355,0.12;
LINE <0.4450003,0.3550003>,<0.4450003,0.3050003>,<0.4550002,0.3050003>,
<0.4350003,0.3050003>,<0.4350003,0.3050003>;
LINE <0.4450003,0.3550003>,<0.4450003,0.4050002>,<0.4550002,0.4050002>,
<0.4350003,0.4050002>;
LINE <0.5450004,0.5650004>,<0.6150002,0.5650004>,<0.6150002,0.5450003>,
<0.5950003,0.540003>,<0.5450004,0.5650004>,<0.5450004,0.5650004>;
LINE <0.4450003,0.3550003>,<0.2850001,0.5150003>;
LINE <0.4500004E-01,0.9100006>,<0.4500004E-01,0.9100006>,
<0.4500004E-01,0.9100006>,<0.4500004E-01,0.9100006>;
LINE <0.4450003,0.3550003>,<0.6550004,0.3550003>,
<0.5850003,0.5450003>,<0.4450003,0.3550003>;
LINE <0.2750002,0.6050004>,<0.2750002,0.5900003>,<0.2850002,0.5900003>,
<0.2900002,0.5950004>,<0.2850002,0.6050004>,<0.2750002,0.6050004>,
<0.2750002,0.6050004>;
POLYGON <0.2650001,0.5550003>,<0.2650001,0.5250002>,
<0.2850001,0.5250002>,<0.2950001,0.5350002>,<0.2850001,0.5550003>,
<0.2650001,0.5550003>;
CIRCLE 0.445,0.355,.031;
ENDPIC;
ENDMF;

```

FIGURE 1 - Exemple de métafichier CGM codé en mode "Texte en clair".

L'interprétation de cette portion du document permet d'obtenir la figure 2.

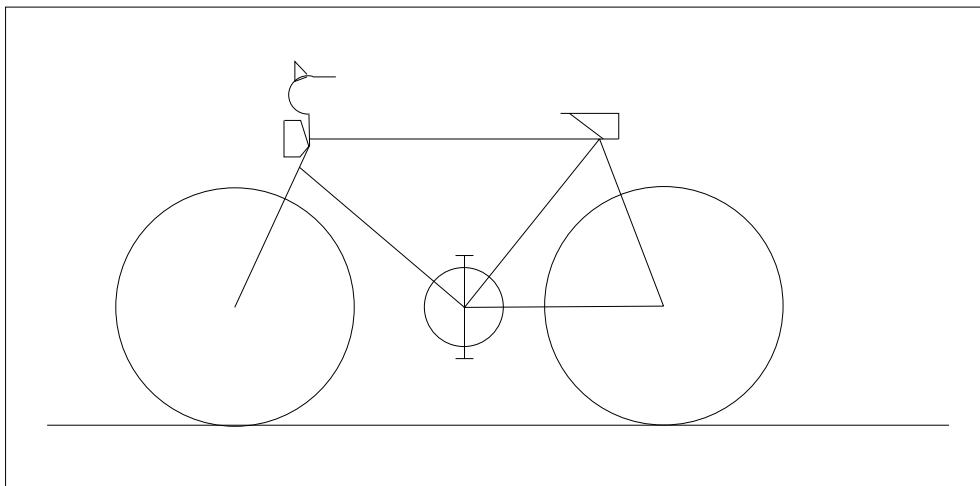


FIGURE 2 - *Illustration correspondant au balisage identifié par FIG2.*

```
<fig id=FIG3 cadre >  
<graphic sizex="80" x0="0.4" y0="0.2" x1="0.8" y1="0.6">  
  BEGMF "Bicycle";  
  ...  
  ENDMF;  
</graphic>  
<titfig> Autre illustration CGM.  
<legfig> Il s'agit en fait d'une autre vue de la même illustration.  
</fig>
```

L'interprétation de ce balisage permet d'obtenir la figure 3.

Si le codage "caractère" est utilisé au lieu du codage "texte en clair", une référence à une entité externe doit être incluse dans le balisage [5]. En effet, comme les délimiteurs SGML peuvent apparaître dans ce fichier, celui-ci doit être incorporé avec un appel de contenu pour qu'il ne soit pas examiné par le parseur SGML. Il peut en toute sécurité être fusionné dans une entité SGML seulement après avoir été examiné et s'il a été établi qu'il ne contenait pas de délimiteurs qui pourrait être connus dans des données textuelles remplaçables, ou s'il a subi un traitement préalable afin de convertir de tel délimiteurs en appels d'entités [1]. Ainsi, si l'illustration en question (figure 1), était contenue dans le fichier "figures.CGM" et que la bicyclette y était représentée par la troisième figure, il faudrait rajouter la ligne suivante à la DTD:

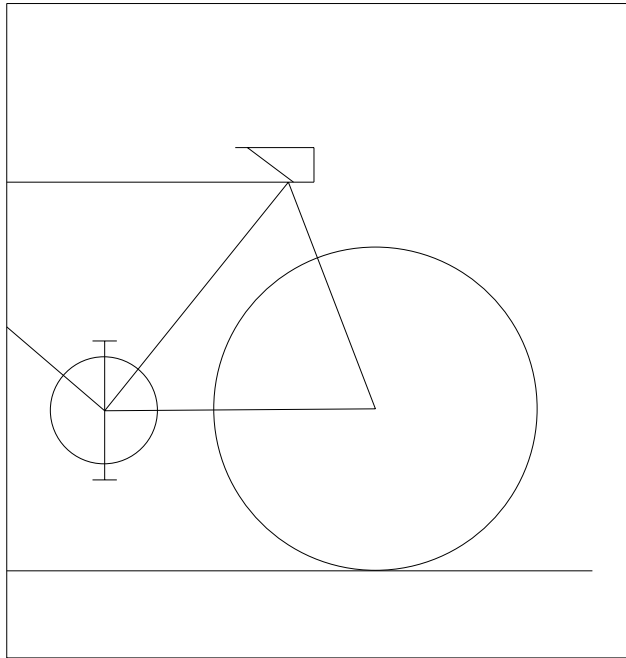


FIGURE 3 - *Illustration correspondant à la portion de document identifiée par FIG3.*

```
<!DOCTYPE ... [  
<!ENTITY pict SYSTEM "figures.CGM" >  
...  
>
```

et faire référence à l'illustration en question par le balisage suivant:

```
<fig cadre >  
<graphic file="pict" cgmchar picnum="3" size="100" >  
<titfig> Figure définie dans un fichier externe.  
<legfig> Ceci représente un autre mode d'utilisation de la norme CGM  
pour l'intégration de graphiques dans un document SGML.  
</fig>
```

8. Génération et interprétation des métafichiers

Bien qu'il puisse être écrit manuellement ou par un éditeur de texte, un métafichier CGM est généralement créé par des outils graphiques dédiés appelés générateurs. Ce métafichier doit par la suite être interprété afin d'afficher son contenu d'images sur un appareil graphique (écran, traceur, etc). Ainsi, le métafichier CGM permet de transmettre des images d'une application de génération à une application d'interprétation, éventuellement distantes dans le temps et dans l'espace, avec des architectures et des ressources pouvant être très différentes.

En effet, CGM est indépendante des systèmes graphiques qui l'utilisent. Les primitives graphiques d'un métafichier définissent des images virtuelles. Les coordonnées de ces éléments (c'est à dire les adresses des points dans l'image virtuelle) sont des coordonnées bi-dimensionnelles absolues de l'espace des coordonnées d'appareil virtuel (VDC). Un appareil virtuel étant un appareil graphique idéal qui comporte un ensemble de fonctionnalités graphiques accessibles aux logiciels graphiques ou aux systèmes par l'intermédiaire de l'interface d'appareil normalisée CGI (Computer Graphics Interface) [7].

L'espace VDC est un espace de coordonnées à deux dimensions dont la précision est infinie et dont la fenêtre est extensible à l'infini. Seul un sous-ensemble de l'espace VDC, le domaine VDC, est réalisable. Ce domaine est composé de toutes les coordonnées représentables dans le format déclaré dans le métafichier.

Pour garantir la portabilité des informations graphiques sur des matériels différents, l'interpréteur doit gérer une interface universelle de commande des appareils définie par la norme CGI. La visualisation effective du contenu d'un métafichier sur un appareil particulier est alors obtenue moyennant les pilotes de cet appareil (driver) qui transforment les commandes CGI en des commandes spécifiques des appareils qu'ils gèrent. Cependant, un interpréteur peut, s'il est nécessaire, être dédié pour un type d'appareil afin d'obtenir une image qui est aussi proche que possible de l'image désirée.

D'un autre côté, l'interpréteur peut être inclus dans une application qui utilise une bibliothèque graphique telle que celle définie par la norme GKS (voir section suivante). On parle dans ce cas de deux niveaux d'interfaces: celui de l'application et celui des appareils. Le métafichier est alors interprété pour gérer la première interface, et c'est la bibliothèque graphique qui se charge par la suite de la visualisation de son contenu graphique à travers la seconde interface. La figure 4 résume les différents cas de génération et d'interprétation des métafichiers CGM en présentant l'architecture générale d'une application graphique.

9. CGM et le GKS

GKS (Graphical Kernel System) est une bibliothèque graphique qui a été définie pour faciliter le développement et l'utilisation d'applications graphiques [9].

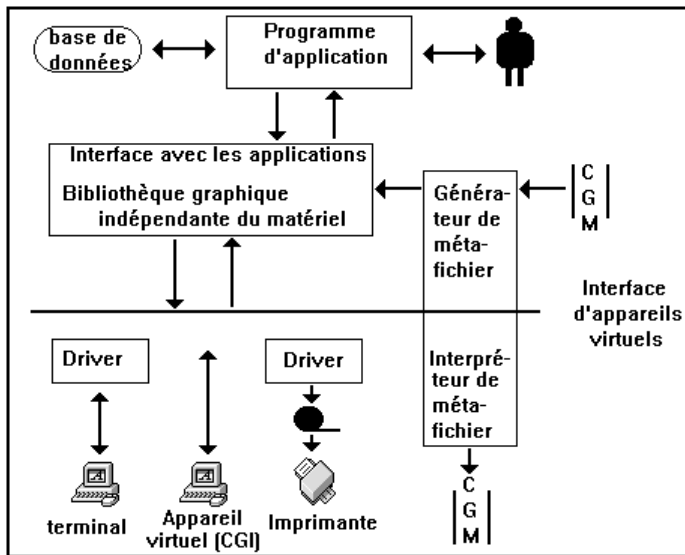


FIGURE 4 - Architecture générale d'une application graphique.

Initialement, elle a été conçue pour offrir les primitives de base pour le traçage en deux dimensions en mode vectoriel et/ou matriciel, ainsi que pour l'interaction avec l'utilisateur (actuellement, il existe une norme internationale qui définit le GKS-3D [9]). La bibliothèque graphique GKS offre au programme d'application une interface graphique soigneusement modélisée qui, par le concept de station de travail, permet l'indépendance du matériel. A chaque périphérique sera associé un programme de contrôle spécifique (driver) qui assurera l'interface avec le noyau du système GKS. Outre la bibliothèque graphique, la norme GKS offre le concept de métafichier. Un métafichier GKS est un fichier séquentiel généré par GKS pour stocker, transporter ou échanger des informations graphiques. Le métafichier s'utilise comme une station de travail virtuelle.

Le concept de métafichier CGM est conforme à celui du GKS. En effet la norme ISO/CEI 8632 tire du système GKS une grande partie des informations de son modèle de système graphique. De plus, elle prescrit un métafichier qui peut être utilisé par le système GKS comme métafichier de capture d'images statiques [6]. La norme ISO/CEI 8632 précise explicitement, d'un autre côté, que la norme de métafichier doit pouvoir être utilisée à partir de systèmes GKS, avec à la fois les fonctions d'entrée (lecture) et de sortie (écriture) de métafichier. La figure 5 représente la relation du GKS avec les autres normes dans la gestion des documents composites [10].

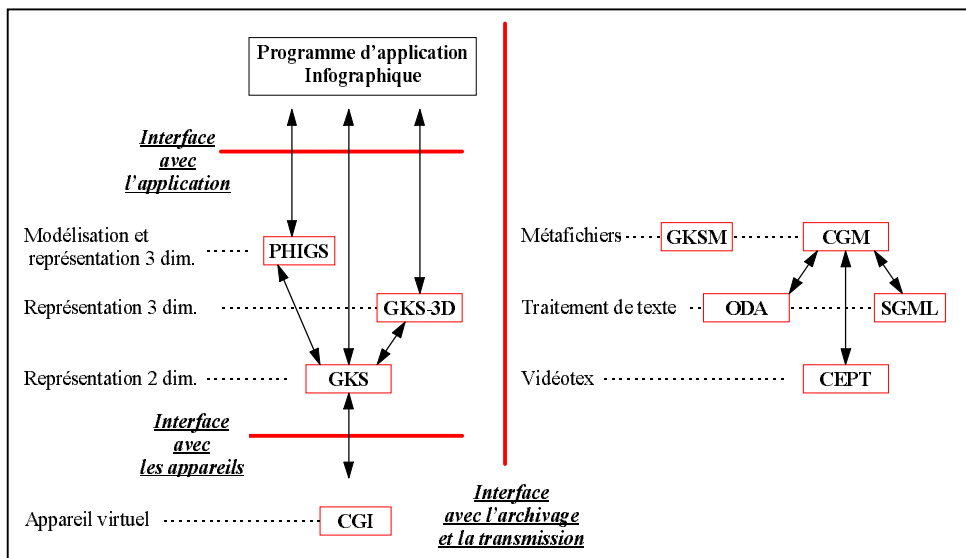


FIGURE 5 - Relation du GKS avec les autres normes dans la gestion des documents composites.

10. Conclusion

L'intégration de graphiques dans un document SGML est donc possible dès que ces graphiques respectent la norme CGM. Deux méthodes peuvent être utilisées pour cette intégration. La première consiste à faire référence dans l'instance de document à un fichier externe normalisé et ce moyennant la définition d'un élément graphique dans la DTD. Le deuxième scénario, permet d'obtenir dans un seul fichier SGML, la totalité d'un document technique contenant aussi bien du texte que des graphiques pour peu que ces derniers soient décrits ou traduits en un ou plusieurs métafichiers CGM complètement conformes et non codés en binaire. Il est à noter, à ce niveau, que dans un même fichier CGM, il est possible de représenter plusieurs figures différentes. Le transfert d'un tel fichier pourra ainsi être assuré en faisant appel au seul format SDIF.

Dans le cadre de nos travaux, nous nous intéressons particulièrement à la gestion électronique des documents techniques multi-normes. L'idée est en fait de concevoir un atelier éditorial multi-normes et multi-plateformes. Cet atelier doit s'appuyer sur une boîte de composants et d'outils (éditeurs, générateurs,...) pouvant être totalement ou partiellement intégrés dans des environnements spécifiques diversifiés. Chaque utilisateur aura la liberté du choix de ses outils logiciels et matériels. Les différentes composantes sont par la suite normalisées et intégrées par le biais d'interfaces normalisées (API) et de passerelles, pour produire

des documents facilement échangeables et retraitables dans des environnements hétérogènes. Dans un premier temps, nous avons focalisé l'attention sur l'aspect relatif à l'intégration des graphiques dans des composants textuels en nous basant sur la norme CGM. Ainsi, ayant constaté que la majorité des logiciels de CAO et de PAO utilisés pour générer les plans et les illustrations des documents techniques, supporte le format de fichier DXF (format propriétaire d'échange défini par AutoDESK), nous avons conçu et développé un outil de conversion de fichiers DXF en métafichiers CGM afin de rendre possible leur intégration dans des documents normalisés. Des travaux en cours se proposent de prendre en considération d'autres normes graphiques, telles que: CCITT-Gr4, IGES, etc.

Bibliographie

- [1] "Traitement de l'information-Systèmes bureautiques-Langage normalisé de balisage généralisé (SGML)", *ISO 8879:1986*;
- [2] "traitement de l'information-facilités de support SGML-Format d'échange de documents SGML (SDIF)", *ISO 9069:1988*;
- [3] "Traitement de l'information-Texte et systèmes bureautiques-ODA et format d'échange", *ISO 8613*
- [4] Danish Standards Association (DS), "SGML-ODA: Présentation des concepts et comparaison fonctionnelle", *coll."Afnor Technique"*, Édition Afnor 1991;
- [5] "traitement de l'information-facilités de support pour SGML-Techniques d'utilisation du SGML", *ISO/IEC TR 9573*, 1988;
- [6] "Technologies de l'information Infographie-Métafichier de stockage et de transfert des informations de description d'images (CGM)", *ISO/CEI 8632-1: Partie 1: Description fonctionnelle, ISO/CEI 8632-2: Partie 2: Codage des caractères, ISO/CEI 8632-3: Partie 3: Codage binaire, ISO/CEI 8632-4: Partie 4: Codage en clair des textes*, 1992;
- [7] "Information technology-Computer graphics-Interfacing techniques for dialogues with graphical devices (CGI)", *ISO/IEC 9636*, 1991;
- [8] "Information processing systems-Computer graphics-Graphic Kernel System (GKS)", *ISO 7942*, 1985;
- [9] "Information processing systems-Computer graphics-Graphic Kernel System for three Dimensions (GKS-3D)", *ISO/IEC 8805*, 1988;
- [10] T. Liebling, H. Rôthlisberger, "Infographie et applications", *Édition Masson*, 1988;