



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.107

(08/95)

TERMINAUX POUR LES SERVICES TÉLÉMATIQUES

**SERVICE D'INTERFACE HOMME MACHINE
AMÉLIORÉE POUR LE VIDÉOTEX ET
AUTRES SERVICES DE CONSULTATION**

Recommandation UIT-T T.107

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1^{er}-12 mars 1993).

La Recommandation UIT-T T.107, que l'on doit à la Commission d'études 8 (1993-1996) de l'UIT-T, a été approuvée le 11 août 1995 selon la procédure définie dans la Résolution n° 1 de la CMNT.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1996

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

		<i>Page</i>
1	Champ d'application.....	1
2	Références.....	1
3	Définitions et abréviations.....	2
	3.1 Définitions.....	2
	3.2 Abréviations.....	3
4	Modèle général.....	4
	4.1 Introduction.....	4
	4.2 Définition des éléments VEMMI.....	4
	4.2.1 Définition et identification d'objet VEMMI.....	5
	4.2.2 Définition de composant VEMMI.....	5
	4.2.3 Définition d'item de composant VEMMI.....	6
	4.2.4 Définition des ressources.....	6
	4.2.5 Transfert d'objets VEMMI.....	6
	4.2.6 Transfert de ressources.....	6
	4.3 Modèle structurel de plan VEMMI.....	7
	4.3.1 Le plan standard pour les données vidéotex.....	7
	4.3.2 Le plan des objets VEMMI.....	7
	4.4 Modes d'exploitation des terminaux VEMMI.....	7
	4.4.1 Le mode standard.....	7
	4.4.2 Le mode VEMMI.....	7
	4.4.3 Commutation entre mode standard et mode VEMMI.....	9
	4.5 Contenu de données des éléments VEMMI.....	9
	4.5.1 Définition des données en mode texte.....	9
	4.5.2 Phototrame.....	10
	4.5.3 Couleurs.....	11
	4.5.4 Données vidéotex.....	11
	4.5.5 Son.....	11
	4.5.6 Données graphiques.....	11
	4.5.7 Données vidéo.....	11
	4.6 Positionnement et dimensionnement des objets VEMMI.....	11
	4.6.1 Positionnement.....	11
	4.6.2 Dimensionnement.....	12
	4.7 Etats et paramètres d'état des éléments VEMMI.....	13
	4.7.1 Objet.....	13
	4.7.2 Composant.....	16
	4.8 Gestion d'action locale.....	17
	4.9 Considérations relatives à la mise en mémoire.....	18
	4.10 Règles communes de traitement des objets.....	18
	4.10.1 Gestion d'état d'activité et de convergence.....	18
	4.10.2 Comportement du mode modal.....	19
	4.10.3 Considérations relatives aux dimensions et au découpage.....	19
	4.11 Mémorisation d'objets locaux.....	19
	4.12 Noms d'annuaire symboliques.....	20
	4.13 Règles propres aux terminaux spécialisés.....	20
5	Description du service.....	21
	5.1 Éléments de service lancés par l'application et le terminal VEMMI.....	23
	5.2 Éléments de service lancés par l'application VEMMI.....	23
	5.2.1 VEMMI_Open (ouverture VEMMI).....	23
	5.2.2 VEMMI_Close (fermeture VEMMI).....	24
	5.2.3 VEMMI_Resume (reprise VEMMI).....	24

5.2.4	VEMMI_Suspend (suspension VEMMI).....	24
5.2.5	VEMMI_Identify_Term_Cap (identification des capacités du terminal VEMMI)	25
5.2.6	VEMMI_Set_Options (sélection d'options VEMMI)	25
5.2.7	VEMMI_Create_Object (création d'objet VEMMI)	26
5.2.8	VEMMI_Open_Object (ouverture d'objet VEMMI)	26
5.2.9	VEMMI_Open_Blocking_Object (ouverture d'objet VEMMI avec blocage)	26
5.2.10	VEMMI_Close_Object (fermeture d'objet VEMMI).....	27
5.2.11	VEMMI_Close_All (fermeture de tous les objets VEMMI).....	27
5.2.12	VEMMI_Destroy_Object (effacement d'objet VEMMI)	27
5.2.13	VEMMI_Obj_Access_Disable (interdiction d'accès à l'objet VEMMI)	27
5.2.14	VEMMI_Obj_Access_Enable (autorisation d'accès à l'objet VEMMI)	28
5.2.15	VEMMI_Modify_Component (modification de composant VEMMI)	28
5.2.16	VEMMI_Obj_Location_Change (modification d'emplacement d'objet VEMMI).....	29
5.2.17	VEMMI_Load_Col_Table (introduction de la table de couleurs VEMMI)	29
5.2.18	VEMMI_Reset_Col_Table (remise à l'état initial de la table de couleurs VEMMI)	30
5.2.19	VEMMI_Open_Application (ouverture d'application VEMMI).....	31
5.2.20	VEMMI_Delete_Outdated_Objects (effacement des objets VEMMI périmés)	31
5.2.21	VEMMI_Store_Objects (mémorisation d'objets VEMMI)	32
5.2.22	VEMMI_Erase_Objects (effacement d'objets VEMMI).....	32
5.2.23	VEMMI_User_Lock (blocage de l'utilisateur VEMMI)	32
5.2.24	VEMMI_User_Unlock (déblocage de l'utilisateur VEMMI).....	32
5.2.25	VEMMI_Resource_Transfer (transfert de ressources VEMMI).....	33
5.3	Eléments de service lancés par le terminal	34
5.3.1	VEMMI_Identify_Term_Cap_Resp (réponse à une demande d'identification des capacités du terminal VEMMI).....	34
5.3.2	VEMMI_Object_Retransmission (retransmission d'objet VEMMI).....	34
5.3.3	VEMMI_User_Data (données d'utilisateur VEMMI)	35
5.3.4	VEMMI_Open_Application_Resp (réponse à une commande d'ouverture d'application VEMMI)	36
5.3.5	VEMMI_Store_Objects_Resp (réponse à une commande de mémorisation d'objets VEMMI)	36
5.3.6	VEMMI_Error (erreur VEMMI)	36
5.3.7	VEMMI_Resource_Transfer_Abort (abandon de transfert de ressources VEMMI)	37
6	Introduction d'objets VEMMI	37
6.1	La barre d'application	38
6.1.1	Composition.....	38
6.2	La barre de boutons	38
6.2.1	Composition.....	38
6.3	Le menu incrusté.....	38
6.3.1	Composition.....	38
6.4	La boîte de dialogue	39
6.4.1	Composition.....	39
6.5	Objet opératif.....	40
6.6	Objet ressource en phototrame	41
6.7	Objet ressource en mode vidéotex	41
6.8	Objet ressource en mode texte	41
6.9	Objet ressource de police.....	41
6.10	Objet de métacode	41
6.11	La boîte de messages	41
7	Description fonctionnelle	41
7.1	Règles générales pour le comportement des éléments	41
7.1.1	Interaction avec l'utilisateur	41
7.1.2	Actions locales et comptes rendus	41
7.1.3	Relations entre objets et composants	42
7.1.4	Ouverture/fermeture d'objets audio, vidéo, ressource et de métacode	43
7.1.5	Opération d'agrandissement plein écran.....	43
7.1.6	Conventions applicables pour les notations	44
7.1.7	Mnémonique	44

	<i>Page</i>	
7.2	Formats de texte.....	44
7.2.1	Texte de haute qualité VEMMI	44
7.2.2	Étiquettes et titres de texte	45
7.3	La barre d'application	46
7.3.1	Composition.....	48
7.4	La barre de boutons	53
7.4.1	Composition.....	54
7.5	Le menu incrusté.....	55
7.5.1	Composition.....	57
7.6	La boîte de dialogue	58
7.6.1	Composition.....	61
7.7	La boîte de messages	83
7.8	Objet opératif.....	85
7.9	Objet de ressource en phototrame.....	87
7.10	Objet de ressource vidéotex	88
7.11	Objet de ressource de texte	88
7.12	Objet de ressource de police	89
7.13	Objet de métacode	89
7.14	Définition du type de données phototrame VEMMI.....	89
7.15	Catalogue identifiant les types de codage de contenu VEMMI	89
8	Représentation codée complète VEMMI	91
8.1	Introduction	91
8.2	Notation utilisée.....	91
8.3	Commutation globale d'environnement de codage	91
8.3.1	Passage en mode VEMMI.....	91
8.3.2	Structure syntaxique ISO/CEI 9281 [14]	93
8.4	Syntaxe de commande VEMMI.....	95
8.5	Objets, composants	98
8.6	Actions locales.....	102
9	Codage	103
9.1	Structure des commandes	103
9.2	Structure des objets, des composants et des attributs.....	103
9.3	Codage des symboles terminaux.....	104
9.3.1	Codes opération	104
9.3.2	Entiers	104
9.3.3	Énumération.....	105
9.3.4	Chaînes	105
9.3.5	NDC	106
9.4	Attributs et symboles de niveau inférieur	107
9.5	Codes opération	108
9.6	Syntaxe de la commande VEMMI_Modify_Component.....	111
9.7	Valeurs par défaut.....	113
10	Introduction du service VEMMI dans les Recommandations existantes concernant le vidéotex.....	116
10.1	Introduction du service VEMMI dans la Recommandation T.101 [4].....	116
10.2	Introduction du service VEMMI dans la Recommandation T.105 [6].....	116
Annexe A	– Chaîne T.51	116
A.1	Champ d'application	116
A.2	Jeux de caractères graphiques.....	116
A.3	Technique d'extension de code	119
A.4	Répertoire du jeu de caractères latins	119
A.5	Fonctions de commande	119
Annexe B	– Jeu secondaire obligatoire de ISO 8859 [13].....	120
Annexe C	– Noyau minimal de types de données.....	121

RÉSUMÉ

La présente Recommandation spécifie la syntaxe à utiliser dans les services vidéotex ou autres services de consultation pour mettre en œuvre une interface homme-machine améliorée pour le vidéotex (VEMMI) (*videotex enhanced man machine interface*).

La VEMMI permet d'améliorer l'ergonomie et l'interactivité des systèmes de consultation par l'utilisation d'éléments graphiques – barre d'application, barre de boutons, menus incrustés, boîte de dialogue, par exemple – dans les dialogues.

Le plan standard continue de recevoir les données standard de la plate-forme de base. Les applications vidéotex standard peuvent donc tourner aussi sur un terminal VEMMI. Le plan des objets VEMMI reçoit les objets VEMMI. Un système de commutation automatique assure le passage en mode standard ou en mode VEMMI.

La présente Recommandation définit des types de données (texte de haute qualité VEMMI, phototrame VEMMI) et autorise l'intégration, dans le dialogue VEMMI, d'autres types de données d'usage courant (par exemple: JPEG, BMP, WAVE, MIDI, Vidéotex). Elle définit également des objets opératifs destinés à élargir les possibilités d'une application VEMMI pendant la durée d'exécution.

Pour améliorer le fonctionnement des applications interactives, on peut répartir les objets VEMMI entre un serveur et un terminal VEMMI. L'application VEMMI exécute les opérations de commande (sauvegarde, chargement, mise à jour) des objets VEMMI dans le terminal.

La présente Recommandation contient la description du service, les éléments du service et leur codage.

SERVICE D'INTERFACE HOMME MACHINE AMÉLIORÉE POUR LE VIDÉOTEX ET AUTRES SERVICES DE CONSULTATION

(Genève, 1995)

1 Champ d'application

La présente Recommandation spécifie la syntaxe de données que les services vidéotex et de consultation d'information multimédia/hypermédia devront utiliser pour mettre en œuvre l'interface homme-machine améliorée pour le vidéotex (VEMMI).

Dans le cas du vidéotex, cette Recommandation est applicable aussi bien au service vidéotex qu'aux terminaux vidéotex associés. Ces terminaux peuvent être raccordés au service vidéotex par l'intermédiaire du réseau téléphonique public commuté (RTPC), du réseau numérique avec intégration des services (RNIS) ou du réseau public pour données à commutation par paquets (RPDCP). De manière générale, les terminaux doivent assurer le service vidéotex syntaxique (SBV) (*syntax-based videotex*) sur le RNIS.

La présente Recommandation peut aussi être utilisée pour tout service de consultation (non apparenté au vidéotex) par l'intermédiaire de la plate-forme de base et des types de données de contenu appropriés.

2 Références

Les Recommandations et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Toute Recommandation ou autre référence est sujette à révision; tous les utilisateurs de la présente Recommandation sont donc invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références indiquées ci-après. Une liste des Recommandations UIT-T en vigueur est publiée régulièrement.

- [1] Recommandation T.50 du CCITT (1992), *Alphabet international de référence (ancien alphabet international n° 5 ou A15) – Technologie de l'information – Jeux de caractères codés à 7 bits pour l'échange d'informations* .
- [2] Recommandation T.51 du CCITT (1992), *Jeux de caractères latins codés pour services de télématique* .
- [3] Recommandation UIT-T T.52 (1993), *Jeux de caractères codés non latins pour les services de télématique* .
- [4] Recommandation UIT-T T.101 (1994), *Interfonctionnement international pour les services vidéotex* .
- [5] Recommandation UIT-T T.102 (1993), *Protocoles de bout en bout pour le vidéotex syntaxique pour le RNIS en mode circuit* .
- [6] Recommandation UIT-T T.105 (1994), *Protocole de couche application pour le vidéotex syntaxique* .
- [7] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuelles à $p \times 64$ kbit/s* .
- [8] Recommandation UIT-T H.320 (1993), *Systèmes et équipements terminaux à bande étroite incluant la visioconférence et la visiophonie* .
- [9] Recommandation UIT-T F.300 (1993), *Service vidéotex* .
- [10] ISO 2022:1986, *Traitement de l'information – Jeux ISO de caractères codés à 7 et à 8 éléments – Techniques d'extension de code* .
- [11] ISO 2375:1991, *Traitement de l'information – Procédure pour l'enregistrement des séquences d'échappement* .
- [12] ISO 8632:1992, *Technologies de l'information – Infographie – Métafichier de stockage et de transfert des informations de description d'images* .
- [13] ISO 8859:1987, *Traitement de l'information – Jeux de caractères graphiques codés sur un seul octet* .
- [14] ISO 9281:1990, *Technologies de l'information – Méthodes de codage d'image* .
- [15] ISO 10646-1:1990, *Technologies de l'information – Jeu universel de caractères codés à plusieurs octets. Partie 1: Architecture et table multilingue* .

- [16] ISO 10918-1:1994, *Technologies de l'information – Compression numérique et codage des images fixes de nature photographique. Partie 1: Prescriptions et lignes.*
- [17] ISO 11172-1:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s. Partie 1: Systèmes.*
- [18] ISO 11172-2:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s. Partie 2: Vidéo.*

3 Définitions et abréviations

3.1 Définitions

Pour les besoins de la présente Recommandation, les définitions suivantes s'appliquent:

- 3.1.1 commandes:** éléments visuels d'interface avec l'utilisateur qui permettent à celui-ci d'entrer en interaction avec des données.
- 3.1.2 zone de visualisation définie (DDA):** voir la Recommandation F.300 [9].
- 3.1.3 accentuation:** surbrillance, modification de couleur ou autre indication visible de l'état d'un élément ou d'une option; effet de cet état sur la capacité d'un utilisateur à entrer en interaction avec cet élément. L'accentuation peut également donner des renseignements supplémentaires sur l'état d'un objet. La méthode utilisée pour accentuer un élément dépend du terminal utilisé.
- 3.1.4 étiquette:** données textuelles associées à un composant VEMMI afin d'informer l'utilisateur de la fonction d'un composant ou item particulier.
- 3.1.5 gestionnaire local:** voir sous gestionnaire local VEMMI.
- 3.1.6 mnémonique:** caractère alphanumérique unique et facile à mémoriser, qui active – et valide – un composant de type option de menu VEMMI. Un caractère mnémonique peut également être utilisé pour valider un poussoir actif.
- 3.1.7 mode modal:** mode caractérisant un objet VEMMI (dit alors «objet modal») que l'utilisateur ne peut pas, au moyen des différents utilitaires d'accès disponibles, quitter au profit d'un autre objet VEMMI de la même application. Toute tentative d'accès de l'utilisateur à un autre objet est vouée à l'échec et éventuellement signalée par un signal acoustique.
- 3.1.8 transfert de ressources:** mécanisme permettant de transférer des fichiers désignés par des objets ressource VEMMI d'une application VEMMI à un terminal VEMMI.
- 3.1.9 présentation à géométrie variable:** affichage en format réduit ou agrandi d'une phototrame, selon la place disponible sur l'écran.
- 3.1.10 présentation en justaposé:** affichage répété d'une phototrame donné dans le sens horizontal et/ou vertical, selon la place disponible sur l'écran.
- 3.1.11 application vidéotex:** application vidéotex utilisant des données, protocoles et profils codés conformément aux Recommandations vidéotex dont les références sont indiquées à l'article 2. Une application vidéotex ne fait pas usage du service, des données et des protocoles VEMMI: voir la Recommandation F.300 [9].
- 3.1.12 données vidéotex:** données échangées entre une application vidéotex et un terminal vidéotex.
- 3.1.13 validation:** action d'activation par un utilisateur, suivie d'une confirmation au clavier ou par dispositif de pointage du choix effectué.
- 3.1.14 application VEMMI:** application offrant une interface homme-machine améliorée comme décrit dans la présente Recommandation.
- 3.1.15 données VEMMI:** description et contenu d'objets VEMMI, ainsi que commandes VEMMI, échangés entre l'application et le terminal VEMMI.
- 3.1.16 gestionnaire local VEMMI:** logiciel exécuté dans le terminal VEMMI afin de traiter et de présenter les objets VEMMI qui sont envoyés à l'utilisateur par l'application VEMMI.
- 3.1.17 terminal VEMMI:** terminal vidéotex qui est en mesure d'exécuter un gestionnaire local VEMMI.
- 3.1.18 terminal vidéotex:** voir la Recommandation F.300 [9].
- 3.1.19 ordinateur principal (serveur) vidéotex:** voir la Recommandation F.300 [9].

3.2 Abréviations

Pour les besoins de la présente Recommandation, les abréviations suivantes sont utilisées:

BIN	Numéro d'identification de phototrame (<i>bitmap identification number</i>)
BMP	Phototrame Microsoft Windows non tributaire du type d'unité (<i>microsoft windows device-independent bitmap</i>)
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CD-ROM	Disque compact ROM (<i>compact disk-read only memory</i>)
CEI	Commission électrotechnique internationale
CGM	Métafichier infographique (<i>computer graphics metafile</i>)
CIN	Numéro d'identification de composant (<i>component identification number</i>)
CMI	Identification de méthode de codage (<i>coding method identifier</i>)
CR	Retour de chariot (<i>carriage return</i>)
DDA	Zone de visualisation définie (<i>defined display area</i>)
DIB	Phototrame non tributaire du type d'unité (<i>device-independent bitmap</i>)
DRCS	Jeu de caractères dynamiquement redéfinissables (<i>dynamically redefinable character set</i>)
DS I	Syntaxe de données conforme à l'Annexe B/T.101 [4]
DS II	Syntaxe de données conforme à l'Annexe C/T.101 [4]
DS III	Syntaxe de données conforme à l'Annexe D/T.101 [4]
ESC	Echappement (<i>escape</i>)
ETS	Norme européenne de télécommunication (<i>european telecommunication standard</i>)
ETSI	Institut européen des normes de télécommunication (<i>european telecommunications standards institute</i>)
FIN	Numéro d'identification de police (<i>font identification number</i>)
G0	Jeu primaire de caractères de la Recommandation T.51 [2]
G2	Jeu supplémentaire de caractères de la Recommandation T.51 [2]
GIF	Format d'échange de graphiques (<i>graphics interchange format</i>)
GMT	Temps moyen de Greenwich (<i>greenwich mean time</i>)
GUI	Interface utilisateur graphique (<i>graphical user interface</i>)
IRV	Version internationale de référence (<i>international reference version</i>)
IS	Norme internationale (<i>international standard</i>)
ISO	Organisation internationale de normalisation (<i>international organisation for standardisation</i>)
JIS	Institut japonais de normalisation (<i>japanese institute for standardization</i>)
JPEG	Groupe d'experts mixte pour la photographie (<i>joint photographic experts groups</i>)
LF	Changement de ligne (<i>line feed</i>)
LI	Indicateur de longueur (<i>length indicator</i>)
MDI	Indicateur de données à suivre (<i>more data indicator</i>)
MIDI	Interface numérique pour instruments de musique (<i>musical instrument digital interface</i>)
MPEG	Groupe d'experts pour les images animées (<i>moving picture experts group</i>)
NDC	Coordonnées normées d'appareil (<i>normalized device coordinate</i>)

OIN	Numéro d'identification d'objet (<i>object identification number</i>)
PCD	Délimiteur de code d'image (<i>picture code delimiter</i>)
PCE	Entité de commande d'image (<i>picture control entity</i>)
PDE	Entité de données d'image (<i>picture data entity</i>)
PE	Élément d'image (<i>picture element</i>)
PI	Identificateur d'image (<i>picture identifier</i>)
PM	Mode image (<i>picture mode</i>)
RGB	Rouge-vert-bleu (<i>red green blue</i>)
RNIS	Réseau numérique avec intégration des services
RPDCP	Réseau public pour données à commutation par paquets
RTPC	Réseau téléphonique public commuté
SBV	Vidéotex syntaxique (<i>syntax based videotex</i>)
TE	Équipement terminal (<i>terminal equipment</i>)
TFI	Identificateur de profil de terminal (<i>terminal facility identifier</i>)
TIN	Numéro d'identification de texte (<i>text identification number</i>)
TLV	Valeur de longueur de type (<i>type length value</i>)
TV	Télévision
UI	Interface utilisateur (<i>user interface</i>)
UIT-T	Secteur de la normalisation des télécommunications de l'Union internationale des télécommunications
VEMMI	Interface homme-machine améliorée pour le vidéotex (<i>videotex enhanced man machine interface</i>)
VIN	Numéro d'identification de mode vidéotex (<i>videotex identification number</i>)
VPDE	Élément de données de présentation vidéotex (<i>videotex presentation data element</i>)
VTX	Vidéotex

4 Modèle général

4.1 Introduction

Entre un serveur et un terminal VEMMI, un service VEMMI traite:

- les objets généraux VEMMI décrits dans la présente Recommandation;
- le contenu de données défini dans la présente Recommandation;
- le contenu de données référencé dans la présente Recommandation.

Un terminal VEMMI peut également assurer une application vidéotex utilisant des données codées et des protocoles tels que décrits dans les Recommandations vidéotex citées à l'article 2.

4.2 Définition des éléments VEMMI

Les unités logiques qui forment la structure VEMMI doivent être désignées et définies comme suit:

- objets VEMMI ou objets;
- composants VEMMI ou composants;
- items de composant VEMMI ou items.

Le terme élément VEMMI est un nom générique qui est utilisé dans la présente Recommandation pour désigner un objet, un composant ou un item.

Un exemple est donné dans la Figure 1.

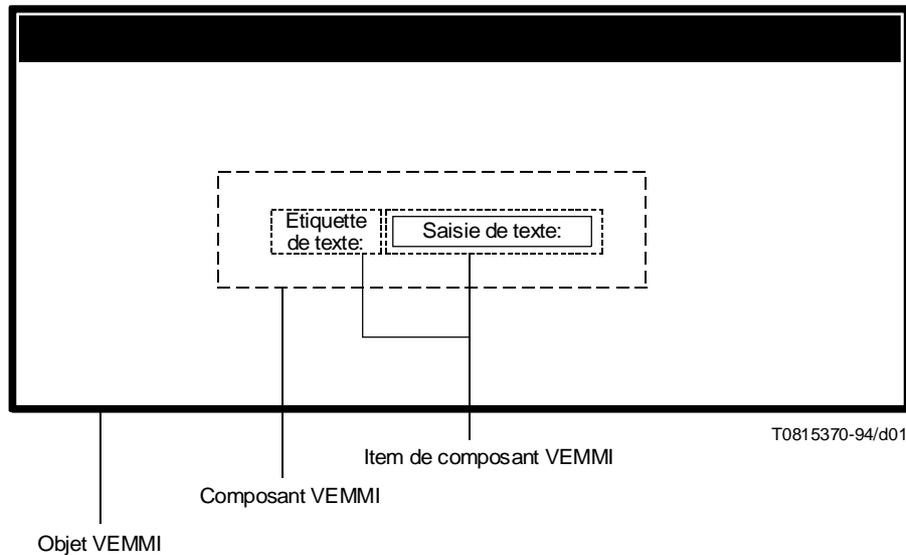


FIGURE 1/T.107

Exemple montrant les objets, les composants et les items

4.2.1 Définition et identification d'objet VEMMI

Les quatre types différents d'objets VEMMI ci-après sont définis dans la présente Recommandation:

- objets visualisés;
- objets opératifs;
- objets ressource;
- objets de métacode.

Sauf indication contraire, le terme objet employé isolément désigne toujours un objet visualisé.

Les objets VEMMI sont les unités logiques qui sont utilisés par une application VEMMI pour assurer l'interactivité avec l'utilisateur.

Les objets VEMMI sont formés de différents composants.

Les objets ne sont définis qu'en ce qui concerne leur fonction, leur taille et leur position par rapport à la zone de visualisation définie (DDA). La représentation des objets dépend du terminal.

Chaque objet doit être identifié par un numéro d'identification d'objet (OIN) qui doit être unique à un moment donné dans une application VEMMI.

4.2.2 Définition de composant VEMMI

Les composants VEMMI font toujours partie d'un objet VEMMI: ils ne sont valides que dans le cadre de cet objet. L'objet auquel un composant appartient est dénommé objet de rattachement.

De façon à transporter des informations, les composants peuvent acheminer un contenu de données (voir 4.5).

Les composants ne sont définis qu'en ce qui concerne leur fonction, leur type de contenu, leur taille et leur position par rapport à l'objet. La représentation des composants dépend du terminal. La représentation du contenu de données est spécifiée dans la présente Recommandation (pour les types de données définis dans la présente Recommandation) ou dans la norme de présentation correspondante (pour les types de données non définis dans la présente Recommandation).

Chaque composant doit être identifié par un numéro d'identification de composant (CIN) qui doit être unique dans le cadre d'un même objet.

4.2.3 Définition d'item de composant VEMMI

Un composant VEMMI est formé d'items de composant. Chaque item fait partie intégrante d'un composant donné. La définition d'un item de composant n'est valide que dans le cadre de ce composant.

4.2.4 Définition des ressources

Les ressources sont des éléments qui peuvent être mentionnés dans les composants ou les objets. Une ressource peut être mentionnée dans plusieurs éléments. On distingue les différentes ressources suivantes:

- la table de couleurs, propre à chaque application;
- les fichiers en mémoire dans le terminal (identifiés chacun par un nom), qui peuvent contenir des données audio, des objets opératifs, etc.;
- une combinaison d'une police et d'un ensemble d'attributs, qui est un objet ressource, identifiée par un numéro d'identification de police (FIN);
- une phototrame, qui est un objet ressource, identifiée par un numéro d'identification de phototrame (BIN);
- un texte, qui peut être un objet ressource, identifié par un numéro d'identification de texte (TIN);
- le mode vidéotex, qui peut être un objet ressource, identifié par un numéro d'identification de mode vidéotex (VIN);
- des ensembles d'objets mémorisés dans le terminal entre deux sessions (identifiés par des attributs).

NOTE – Les numéros FIN, BIN et VIN sont des numéros d'identification d'objet (OIN). Les abréviations FIN, BIN et VIN sont utilisées uniquement pour préciser que l'objet correspondant est un objet ressource.

4.2.5 Transfert d'objets VEMMI

Les objets VEMMI peuvent être transmis au terminal VEMMI par l'intermédiaire d'un réseau de télécommunication. S'ils sont mémorisés à ce stade dans le terminal, ils deviennent des objets locaux. Les objets VEMMI peuvent aussi être transférés à l'occasion de tout transfert de fichiers. Ils deviennent eux aussi des objets locaux et sont traités de la même manière que des objets transférés pendant le dialogue VEMMI et mémorisés par la primitive de service correspondante. Les objets VEMMI peuvent également être transmis au terminal VEMMI par la poste (CD-ROM, disquettes, etc.).

4.2.6 Transfert de ressources

Les objets ressources VEMMI peuvent mentionner des fichiers contenant les données d'affichage des ressources. Ces fichiers sont appelés fichiers de données de ressources.

VEMMI précise le moyen utilisé pour transmettre les fichiers de données de ressources au terminal (transfert de ressources VEMMI). Les fichiers de données de ressources VEMMI peuvent également être transmis au terminal VEMMI selon la procédure de transfert de fichiers utilisée dans le service standard dans lequel fonctionne le terminal ou par la poste (CD-ROM, disquettes, etc.).

Pour offrir un niveau de qualité satisfaisant et des caractéristiques correspondant à la fonctionnalité VEMMI, le mécanisme de transfert des ressources offre les particularités suivantes:

- le transfert des ressources peut être assuré sans que l'utilisateur en soit informé;
- le transfert des ressources peut être assuré en même temps qu'une autre tâche. Pendant le transfert des ressources, l'utilisateur peut poursuivre son interaction avec l'application VEMMI (encore que si la vitesse de transmission sur le réseau est insuffisante, le temps de réponse du serveur risque d'être défavorablement affecté par l'opération de transfert des ressources);
- l'application peut ouvrir un objet VEMMI (fenêtre) indiquant l'état d'avancement du transfert en cours (par exemple un graphique indiquant le pourcentage des ressources copiées), et, sur option, offrir une méthode pour annuler, suspendre et reprendre le transfert (à l'aide de boutons par exemple);
- plusieurs transferts de ressources peuvent être assurés indépendamment les uns des autres;
- interaction possible avec l'utilisateur sur le fonctionnement du transfert des ressources (abandon).

La prise en charge des ressources est facultative pour un terminal VEMMI. S'il prend en charge la fonction d'enregistrement local, le terminal doit aussi prendre en charge le transfert des ressources VEMMI.

4.3 Modèle structurel de plan VEMMI

Le modèle d'affichage VEMMI comprend deux plans indépendants:

- la mémoire de données standard et la fenêtre de données standard;
- la mémoire de données VEMMI et la fenêtre de données VEMMI.

Ce modèle et une structure possible de terminal sont présentés à la Figure 2.

Un terminal VEMMI doit appliquer le comportement de ce modèle d'affichage. Aucune hypothèse n'est cependant avancée pour ce qui est de la structure réelle du plan physique du terminal et de la manière dont celui-ci met en œuvre ce modèle structurel de plan.

Le plan standard est la zone d'affichage où sont présentées les données du service de consultation faisant office de plate-forme pour le service VEMMI. En cas d'utilisation d'un service vidéotex comme plate-forme, le plan logique standard est la DDA qu'utilise une application vidéotex ordinaire. Dans les paragraphes qui suivent, on suppose que le vidéotex est utilisé comme plate-forme. Toutefois, les règles applicables à une plate-forme de base autre que le vidéotex (VT100, canal TV, par exemple) sont rigoureusement identiques.

Les deux plans d'affichage sont indépendants par rapport à leurs positions et dimensions sur l'écran du terminal.

4.3.1 Le plan standard pour les données vidéotex

Ce plan reçoit les données vidéotex standard. Il doit continuer à observer, au sujet des données vidéotex, les règles et priorités définies dans les Recommandations vidéotex citées à l'article 2.

4.3.2 Le plan des objets VEMMI

Ce plan reçoit les objets VEMMI qui sont décrits dans la présente Recommandation, avec leur contenu de données codées. Le plan des objets VEMMI doit être en mesure de gérer le chevauchement d'objets et d'assurer les mécanismes de rétablissement cités dans 4.6 et 4.10.1. A l'intérieur de ce plan et pour un contenu de données déterminé, les règles et priorités définies dans les normes correspondantes sont applicables aux données.

4.4 Modes d'exploitation des terminaux VEMMI

Quel que soit le mode d'exploitation qu'il utilise, le terminal doit toujours être doté d'outils propres à gérer les fonctions spécifiques de la plate-forme (déconnexion, etc.).

4.4.1 Le mode standard

C'est le mode d'exploitation initial d'un terminal VEMMI lors de sa mise en marche ou de sa réinitialisation.

En mode standard, le terminal VEMMI affiche les données standard dans le plan standard. Ce type de données standard dépend du service de consultation auquel le terminal VEMMI est connecté. Les données standard sont entièrement visibles. Les saisies de l'utilisateur ne sont pas contrôlées par le gestionnaire local VEMMI.

Le terminal doit ignorer toutes les commandes VEMMI sauf VEMMI_Open ou VEMMI_Resume si la session VEMMI précédente a été suspendue par une commande VEMMI_Suspend. Cette commande doit effectuer une commutation sur le mode VEMMI. (Les commutateurs d'activation/désactivation VEMMI sont alors mis en position d'activation.)

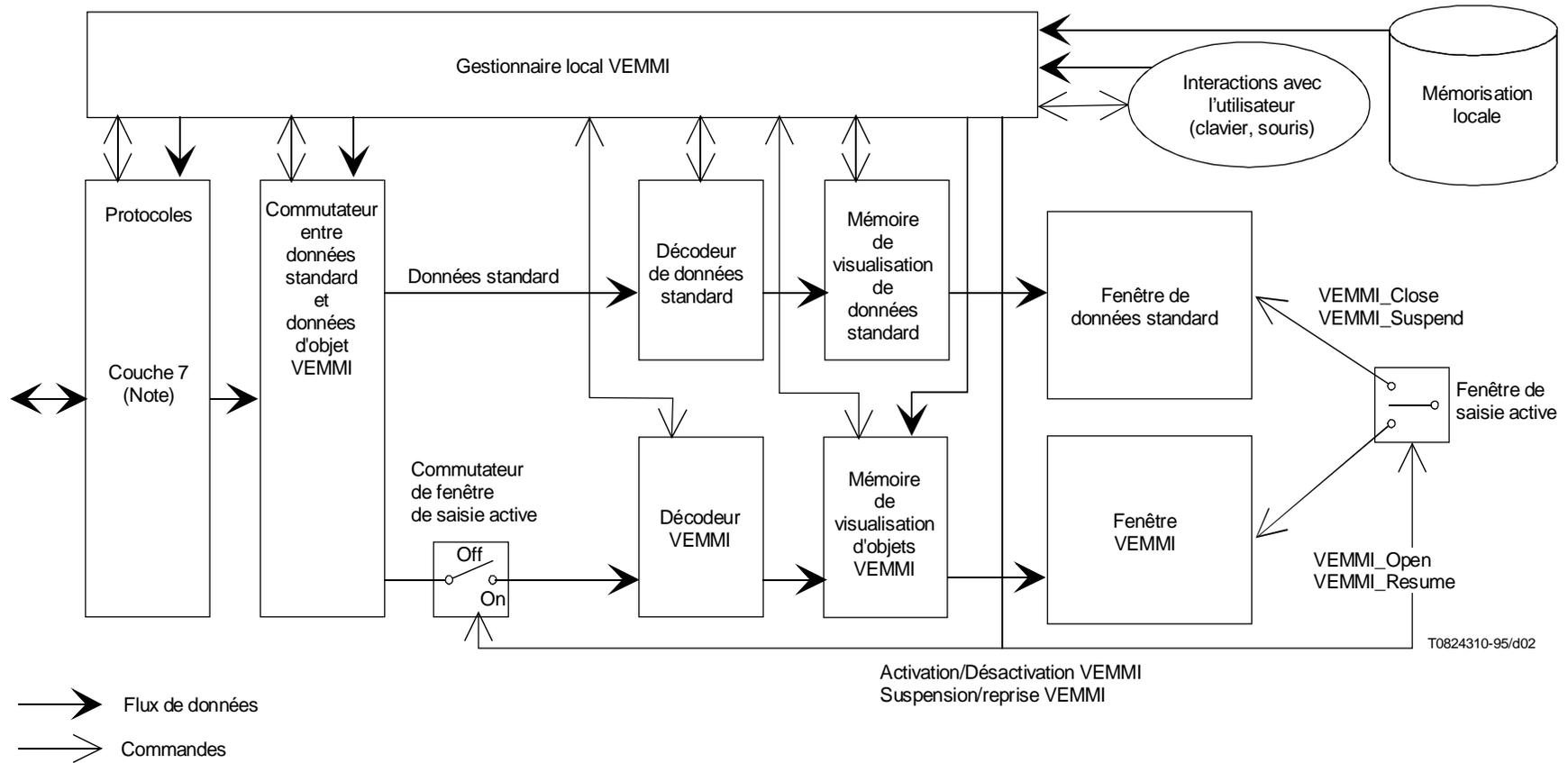
4.4.2 Le mode VEMMI

En mode VEMMI, un terminal VEMMI peut recevoir des données standard et des données d'objets VEMMI selon deux trajets parallèles, comme indiqué sur la Figure 2.

Les données d'objets VEMMI sont affichées dans le plan des objets VEMMI. Les données standard sont affichées dans le plan standard. L'ordre d'affichage des objets et de leurs composants doit correspondre à l'ordre croissant de leurs CIN (le composant ayant le plus petit CIN étant affiché en premier).

En mode VEMMI, les données standard reçues servent à mettre à jour la mémoire d'affichage des données standard. Une mise à jour de la fenêtre des données standard ne déclenche pas l'activation de celle-ci.

Les saisies de l'utilisateur sont contrôlées par le gestionnaire local VEMMI, sur la fenêtre VEMMI, qui est la fenêtre active dans ce mode.



NOTE – Protocoles standard, y compris les protocoles vidéotex syntaxiques (SBV) ou les protocoles modifiés pour le mode VEMMI comme défini à l'article 5.

FIGURE 2/T.107
Exemple de structure possible de terminal VEMMI

4.4.3 Commutation entre mode standard et mode VEMMI

Pour faire passer un terminal VEMMI du mode standard au mode VEMMI, il faut utiliser la commande VEMMI_Open ou VEMMI_Resume.

La commande VEMMI_Open permet de faire passer le terminal du mode initial au mode interface VEMMI.

Lorsqu'une commande VEMMI_Close est reçue, le terminal VEMMI doit passer en mode d'exploitation standard.

Les commandes VEMMI_Resume et VEMMI_Suspend permettent de passer temporairement d'un mode à l'autre.

4.5 Contenu de données des éléments VEMMI

Les types de données de contenu et attributs ci-dessous sont définis dans la présente Recommandation:

- a) texte de haute qualité VEMMI;
- b) phototrame VEMMI;
- c) couleur.

Les types de données de contenu et attributs ci-dessous sont définis ailleurs que dans la présente Recommandation; ils y sont toutefois utilisés comme contenu de données pour les éléments VEMMI:

- a) données vidéotex (sous forme de contenu de données facultatif de la zone d'affichage de graphiques);
- b) phototrame (JPEG, GIF, BMP);
- c) son (WAVE, MIDI);
- d) données graphiques;
- e) données vidéo.

Il est obligatoire pour un terminal VEMMI de comprendre toutes les commandes liées aux contenus de données susmentionnés.

Aucun type de données de contenu particulier n'est obligatoire dans une application VEMMI. Toutefois, pour assurer un service VEMMI satisfaisant, un terminal doit au moins pouvoir accepter les types de données répertoriés dans l'Annexe C.

4.5.1 Définition des données en mode texte

Les données en mode texte sont codées conformément à ISO 8859 [13]. Selon la version du terminal, il se peut qu'une partie seulement de la série 8859 soit acceptée (voir l'Annexe B pour le jeu secondaire obligatoire de ISO 8859). En outre, l'application assurée par le serveur peut utiliser la commande VEMMI_Identify_Term_Cap pour obtenir des renseignements sur les jeux de caractères acceptés dans le terminal. La commande VEMMI_Set_Options permet de sélectionner un jeu de caractères spécifique. Ce codage des données en mode texte est considéré comme le codage de base dans ce mode. Lorsque plusieurs jeux de caractères sont utilisés dans la même application (application multilingue), celle-ci peut utiliser les séquences d'échappement de désignation appropriées homologuées par l'ISO, telles que définies dans ISO 2022 [10] et ISO 8859 [13], pour passer d'un jeu de caractères à un autre.

Les formes de codage des données en mode texte suivantes peuvent être utilisées en options:

- ISO 10646-1 Unicode [15];
- Recommandation T.51 [2], telle qu'elle figure dans l'Annexe A;
- Recommandation T.52 [3].

Pour afficher des titres dans des fenêtres, des options de menu et des étiquettes, l'application du terminal peut utiliser la police du système local. Pour l'affichage de données en mode texte propres à l'application, VEMMI offre des possibilités améliorées. Toutefois, les terminaux qui utilisent un texte simple peuvent recourir à la police du système pour afficher le texte de haute qualité. Ces terminaux peuvent ne pas tenir compte des attributs en mode texte qu'ils ne sont pas en mesure de traiter et afficher le texte à l'aide de la police du système, sur la zone de fond (arrière plan) de celui-ci, dans la couleur disponible.

Les données en mode texte peuvent faire partie de la définition d'un objet ou d'un composant ou encore être définies comme étant des objets ressources. On distingue les attributs textuels suivants:

- couleur du texte;
- police du texte;
- hauteur du texte.

La **couleur du texte** définit la couleur de la zone de front (premier plan) des caractères. Le fond (arrière plan) des caractères est propre au terminal utilisé. La **police du texte** est un jeu de caractères d'un modèle particulier. Dans ce contexte, on peut choisir pour une application une famille de polices, le choix d'une police particulière à l'intérieur de cette famille étant fonction de l'application du terminal (dans la présente Recommandation, le terme «police» désigne une famille de polices). Les familles de polices VEMMI sont présentées dans le Tableau 1.

TABLEAU 1/T.107

Familles de polices VEMMI

Famille de polices	Caractéristiques de police	Exemple
SWISS	Proportionnelle, sans empâtement	Helvetica, suisse
ROMAN	Proportionnelle, avec empâtement	Times roman
FIXFONT	Chasse fixe	Courier

Les polices de ces familles doivent pouvoir accepter des tailles variables (polices réglables).

La **hauteur du texte** précise la hauteur de la police. Il s'agit de la hauteur de la case du caractère, y compris l'interlignage interne éventuel. Le gestionnaire local doit insérer un espace supplémentaire entre rangées de texte adjacentes. La hauteur se mesure en points (1/72 pouce). Un point correspond à 1/400 NDC. L'attribut **Italic** permet de mettre en italique les caractères. L'attribut **Bold** (gras) augmente l'épaisseur des traits et l'attribut **Underline** (soulignement) trace une ligne sous chaque caractère.

Exemple:

SWISS, 10 points et gras

Roman, 16 points, italique

Roman, 8 points et souligné

FIXFONT, 12 points, blanc

Par ailleurs, une combinaison d'une police et d'un ensemble d'attributs, dénommée police attribuée, est traitée comme un objet ressource de police. Une telle combinaison est désignée par le numéro d'identification de police (FIN).

Pour présenter le texte correctement, il convient d'utiliser les caractères de commande suivants:

- SP, NBSP, SHY: voir ISO 8859-1 [13];
- CR, LF: voir la Recommandation T.50 [1].

Pour assurer la fonction «interligne» dans le contenu en mode texte d'un composant ou d'un item de composant, il convient d'utiliser les caractères de commande retour chariot (CR) + interligne (LF) indiqués dans la Recommandation T.50 [1].

4.5.2 Phototrame

Une phototrame est une matrice de pixels qui contient des indices de couleur, qui renvoient à la table de couleurs, ou des composants rouge-vert-bleu (RGB). Une phototrame peut être désignée par un objet ou un composant et dans le cas où elle est instanciée par les terminaux elle se présente sous la forme d'un rectangle de même structure chromatique que la matrice de pixels correspondante. La définition est presque indépendante de l'appareil utilisé, alors que la relation entre les bits de la phototrame et les pixels apparaissant sur l'appareil est propre à chaque appareil.

La phototrame est traitée comme un objet ressource et désignée par des objets ou des composants par l'intermédiaire du «numéro d'identification de phototrame» (BIN), qui est unique dans une application donnée. Plusieurs phototrames peuvent être définies à un moment donné.

4.5.3 Couleurs

La table de couleurs offre une méthode pour l'accès aux fonctions couleur de l'équipement terminal. On admet que celui-ci peut afficher au moins 256 couleurs simultanément. La table de couleurs de l'équipement terminal étant commune à plusieurs applications, les interfaces utilisateur (UI) prévoient des mécanismes (palettes de couleurs logiques, par exemple) permettant d'attribuer à chaque application sa propre table. La table de couleurs définie dans le présent paragraphe est celle de l'application VEMMI.

Sur les 256 couleurs, l'interface utilisateur (UI) en réserve 20 pour l'établissement d'une table du système. Le nombre maximal de couleurs de la table de couleurs VEMMI est donc de 236. Dans la pratique, une application n'utilisera qu'un petit nombre de ces couleurs. L'application assurée par le serveur permet de définir la table de couleurs, par l'envoi des composantes RGB. Des objets, des composants ou des phototrame peuvent choisir ces couleurs au moyen d'indices. Par défaut, 16 valeurs de couleur (de 0 à 15) sont prédéfinies. Si l'application utilise un indice de couleur non défini > 15, la couleur affichée dépend du terminal utilisé.

Les valeurs de couleur peuvent être utilisées par d'autres applications actives dans les terminaux. Il appartient à l'interface utilisateur locale d'assurer la gestion de ces opérations.

Les terminaux noir et blanc admettent deux indices de couleur (par exemple: noir et blanc ou la table de couleurs étoffée des nuances d'une couleur).

4.5.4 Données vidéotex

Données codées conformément à l'Annexe B/T.101, l'Annexe C/T.101 et l'Annexe C/T.101 [4], pour les textes et les figures géométriques, par exemple.

Données codées conformément à l'Annexe E/T.101 [4], pour les données audio.

Données codées conformément à l'Annexe F/T.101 [4], pour les données photographiques.

L'admission de données vidéotex est facultative pour un terminal VEMMI.

La commande VEMMI_Identify_Term_Cap permet à l'application assurée par le serveur de demander si ces types de données sont admis.

4.5.5 Son

Une application peut intégrer des rappels de séquences sonores à l'intérieur d'objets. Le terminal enregistre le son dans des fichiers. L'utilisateur déclenche par une action le traitement du son, éventuellement au moyen d'une carte d'interface audio.

Les deux formats sonores sont adaptés aux équipements suivants:

- équipements WAVE;
- équipements d'interface numériques pour instruments de musique (MIDI).

On peut aussi produire le son à l'aide de la syntaxe audio vidéotex définie en tant que contenu de données vidéotex (voir 4.5.4).

4.5.6 Données graphiques

La présente Recommandation ne prévoit pas la prise en charge directe des données graphiques telles que les lignes, les arcs, les zones de remplissage et les attributs de dessin correspondants. Elle prévoit toutefois une application permettant d'utiliser des programmes intégrés à VEMMI (par l'intermédiaire d'objets opératifs), assurant l'exécution d'opérations graphiques.

4.5.7 Données vidéo

La présente Recommandation ne décrit pas les formats de données vidéo. Toutefois, il fournit le conteneur (objet) qui peut être utilisé pour intégrer la vidéo dans les applications VEMMI.

4.6 Positionnement et dimensionnement des objets VEMMI

4.6.1 Positionnement

Le plan standard doit continuer à supporter le système de coordonnées utilisé dans la plate-forme de base.

Le plan logique des objets VEMMI doit supporter un système de coordonnées normées d'appareil (NDC) pour le positionnement des objets et composants VEMMI à l'intérieur de la DDA (voir la Figure 3). Les coordonnées normées sont théoriquement exprimées par des valeurs comprises entre 0;0 et 1;1. Le point de référence 0;0 indique l'origine des coordonnées et correspond au coin supérieur gauche de la DDA. Le point de coordonnées 1;1 correspond au coin inférieur droit d'une DDA théoriquement carrée, dont le côté est égal à l'unité. Pour une DDA non carrée, l'unité est égale au plus grand côté de la DDA.

Pour les dispositifs d'affichage types, utilisant le format habituel 4:3, le positionnement horizontal s'effectue entre 0 et 1, correspondant à la pleine largeur de la DDA. Le positionnement vertical s'effectue entre 0 et 0,75, correspondant à la pleine hauteur de la DDA.

Le point de référence origine localisant un objet dans la DDA est toujours le coin supérieur gauche de cette zone. Le point de référence origine localisant un composant à l'intérieur de son objet de rattachement (objet conteneur) est également le coin supérieur gauche d'une DDA virtuelle rattachée au coin supérieur gauche de l'objet. Cette DDA virtuelle a exactement la même taille que la DDA standard, mais avec une translation de position jusqu'au point de référence origine de l'objet dont elle dépend. Chaque objet VEMMI possède donc une DDA virtuelle pour positionner ses propres composants.

Les items des composants VEMMI sont généralement positionnés de manière implicite par rapport à la largeur ou à la hauteur du plus proche item situé à gauche ou au-dessus.

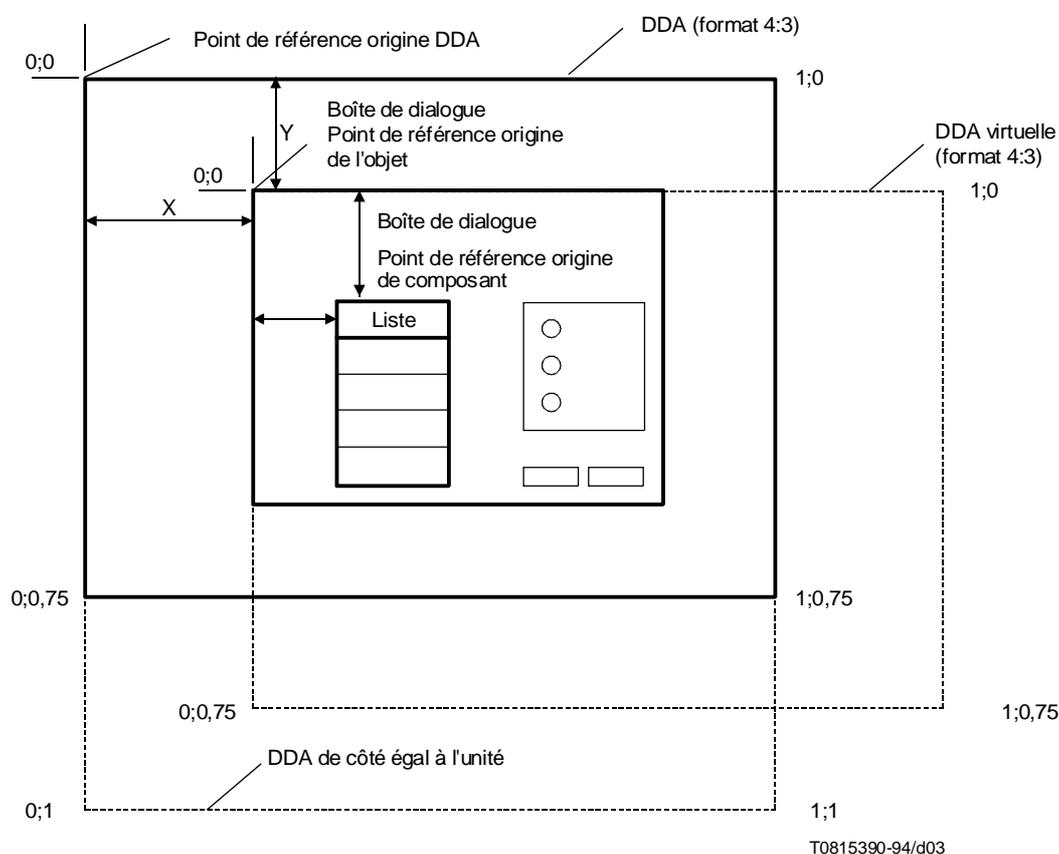


FIGURE 3/T.107

Espace en NDC pour le positionnement VEMMI

4.6.2 Dimensionnement

Le plan standard doit continuer à supporter le système de coordonnées utilisé pour le dimensionnement dans la plateforme de base.

Le plan des objets VEMMI ainsi que les objets et composants VEMMI doivent supporter un système de dimensionnement en NDC.

La largeur et la hauteur d'un objet ou composant VEMMI, exprimées en NDC, sont exprimées par rapport à leur point de référence de positionnement (coin supérieur gauche) (voir la Figure 4).

Dans un élément VEMMI, un contenu de données suit les règles de codage indiquées dans la syntaxe de données qui le concerne, y compris une éventuelle insertion ou mise en page d'image à l'aide du système de coordonnées correspondant.

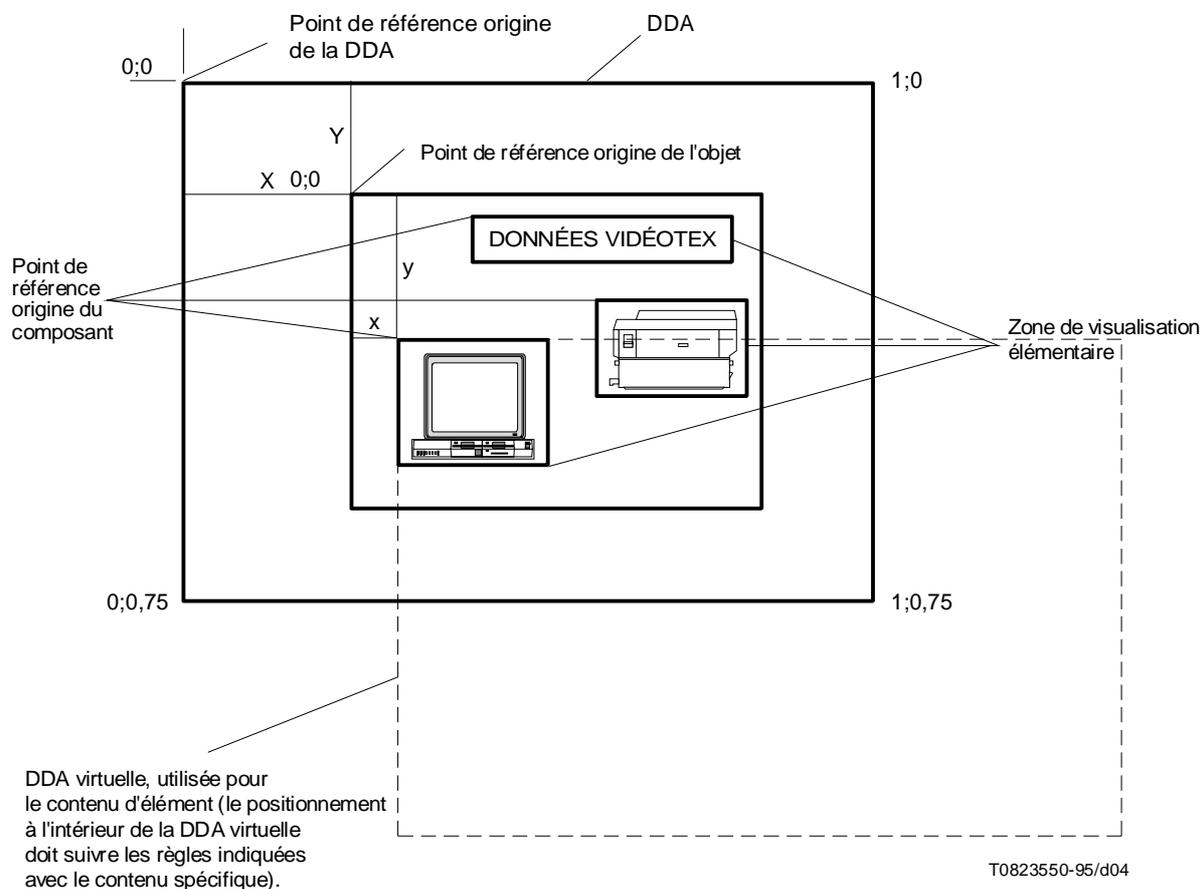


FIGURE 4/T.107
Système de positionnement VEMMI

4.7 Etats et paramètres d'état des éléments VEMMI

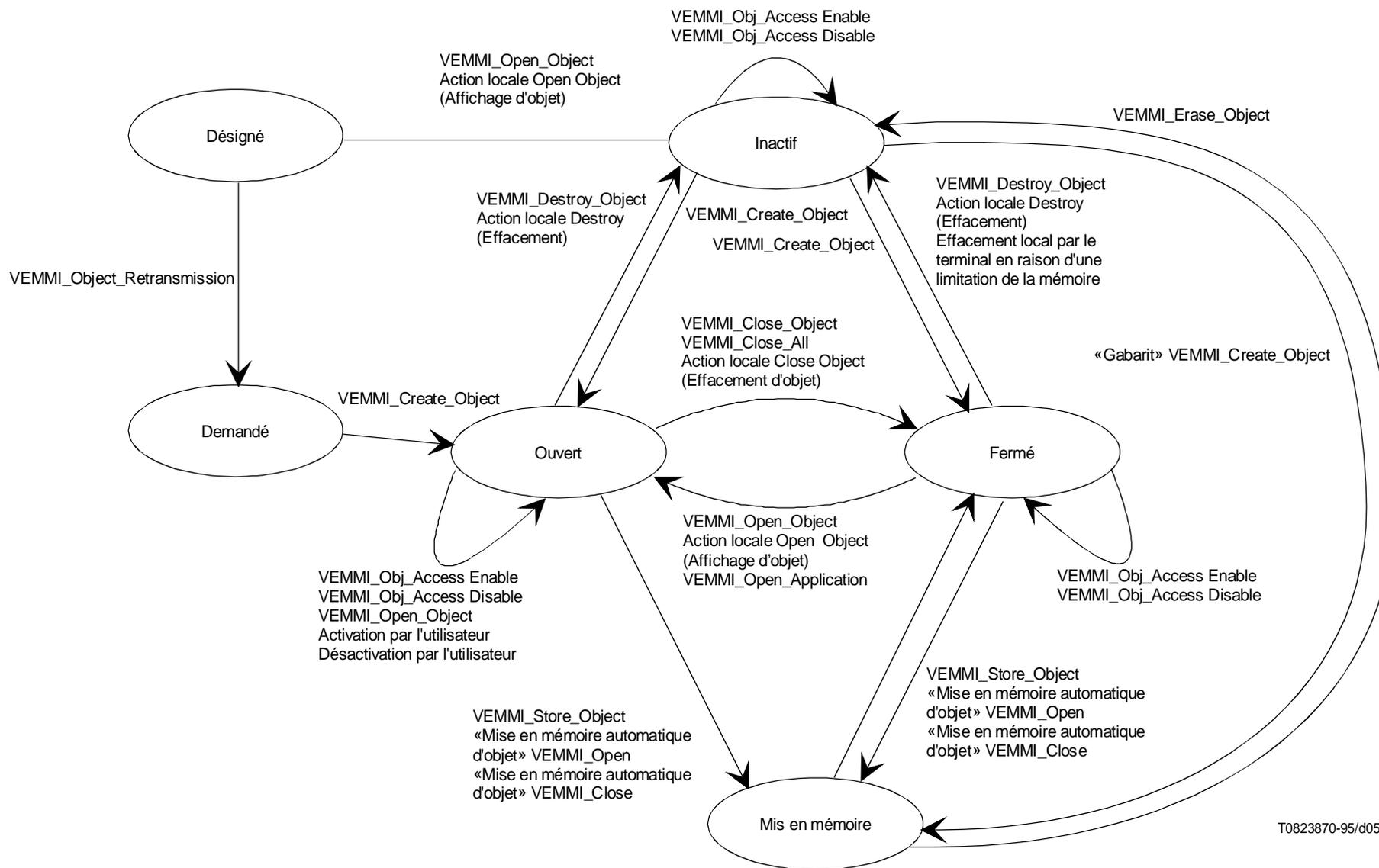
4.7.1 Objet

Un objet peut adopter différents états logiques avec différents paramètres d'état. Ces états et paramètres d'état peuvent résulter:

- de saisies par l'utilisateur;
- d'actions locales;
- d'actions lancées par le gestionnaire local VEMMI;
- de commandes émanant de l'application VEMMI, directement ou par l'intermédiaire d'un objet de métacode.

4.7.1.1 Définition des états d'objet

La Figure 5 montre les commandes qui peuvent être utilisées par l'application VEMMI pour modifier l'état ou les paramètres d'état d'un objet.



T0823870-95/d05

FIGURE 5/T.107
Diagramme d'état pour les objets

En général, les états logiques suivants sont définis pour les objets, avec certaines restrictions en ce qui concerne des objets spécifiques (voir l'article 7):

- Etat *ouvert*:

dans cet état, l'objet existe dans le terminal et est affiché.

L'ouverture d'un objet est lancée par l'application VEMMI, soit en tant qu'état initial de l'objet au moment de sa création, par commande VEMMI directe à un moment quelconque de l'application ou par une action locale associée à un composant et déclenchée par l'interaction avec l'utilisateur.

- Etat *fermé*:

dans cet état, l'objet existe dans le terminal et n'est pas affiché.

La fermeture d'un objet est lancée par l'application VEMMI, soit en tant qu'état initial de l'objet au moment de sa création par commande à un moment quelconque de l'application ou par une action locale associée à un composant et déclenchée par l'interaction avec l'utilisateur.

- Etat *désigné*:

dans cet état, un objet qui n'existe pas dans le terminal a été désigné par une commande VEMMI_Open_Object ou l'action locale «Open object»: Le terminal doit demander la retransmission de l'objet spécifié au moyen de la commande VEMMI_Object_Retransmission.

- Etat *demandé*:

à l'issue de la tentative d'ouverture d'un objet qui n'existe pas dans le terminal, celui-ci envoie à l'application VEMMI une demande de retransmission d'objet. Pendant que le terminal attend que l'objet demandé soit créé, celui-ci est à l'état demandé.

- Etat *mémorisé*:

copie d'un objet (avec tous ses composants), qui a été utilisé au cours d'une session antérieure ou de la session effectivement ouverte et qui a été enregistré dans la mémoire permanente du terminal (sur disque dur, par exemple).

Un objet mémorisé doit implicitement se trouver à l'état fermé. La commande d'ouverture d'application (Open Application) rend les objets mémorisés disponibles pour des opérations sur les objets. Les objets mémorisés se comportent alors comme des objets fermés reçus.

Si un objet déjà mémorisé dans le terminal fait l'objet d'une nouvelle demande de mise en mémoire par l'application assurée par le serveur, le nouvel objet remplacera l'ancien avec tous ses paramètres et états.

Les objets mémorisés peuvent être effacés à l'aide de la commande VEMMI_Erase_Objects.

Un objet qui se trouve à l'état fermé peut être effacé de la mémoire du terminal sur décision du gestionnaire local VEMMI (par exemple en raison d'une insuffisance de capacité). L'application VEMMI n'est pas informée de cet effacement local utilisant la commande d'erreur VEMMI «Object Destroy Indication» (indication d'effacement d'objet). Si le terminal reçoit une commande VEMMI_Open_Object pour un objet qui a été effacé ou si une action locale «Open object» concernant un objet effacé est exécutée, le terminal doit demander la retransmission de l'objet par l'application, au moyen de la commande VEMMI_Object_Retransmission. L'application VEMMI doit alors recréer l'objet demandé au moyen de la commande VEMMI_Create_Object, en utilisant l'état actuel et les paramètres d'état actuels de cet objet. Une commande VEMMI_Create_Object consécutive à une demande VEMMI_Object_Retransmission doit toujours donner lieu à la création d'un objet à l'état ouvert.

4.7.1.2 Définition des paramètres d'état des objets

En général, les paramètres d'état suivants sont définis pour les objets, avec certaines restrictions en ce qui concerne des objets spécifiques (voir l'article 7):

- Etat *actif*:

dans cet état, un objet est ce vers quoi la saisie converge actuellement. Les saisies d'utilisateur se rapportent toujours à l'objet actif. Celui-ci se situe dans le haut de la DDA. Si l'objet actif est inaccessible, l'utilisateur ne peut entrer en interaction avec lui.

La gestion de l'état d'activité est décrite au 4.10.1.

- Etat *inactif*:

état inverse de l'état actif.

- Etat *accessible*:

dans cet état, l'utilisateur peut entrer en interaction avec l'objet.

C'est l'application VEMMI qui permet l'entrée en interaction avec un objet, soit en tant qu'état initial au moment de la création de cet objet, par une commande VEMMI à n'importe quel moment de l'application ou par une action locale associée à un composant et déclenchée par l'interaction avec l'utilisateur.

- Etat *inaccessible*:

état inverse de l'état accessible.

Le paramètre d'état actif/inactif ne s'applique qu'aux composants qui sont à l'état ouvert. Le paramètre d'état accessible/inaccessible s'applique aux composants qui sont à l'état ouvert et à ceux qui sont à l'état fermé. Les paramètres d'état actif/inactif et accessible/inaccessible peuvent se combiner de n'importe quelle manière pour un composant à l'état ouvert.

4.7.2 Composant

4.7.2.1 Définition des états de composant

Un composant peut adopter différents états logiques avec différents paramètres d'état. Ces états et paramètres d'état peuvent résulter:

- de saisies par l'utilisateur;
- d'action locales;
- de commandes émanant de l'application VEMMI.

La Figure 6 montre les états que les composants peuvent adopter. Les états et les paramètres d'état des composants peuvent être modifiés au cours des applications au moyen de la commande VEMMI_Modify_Component.

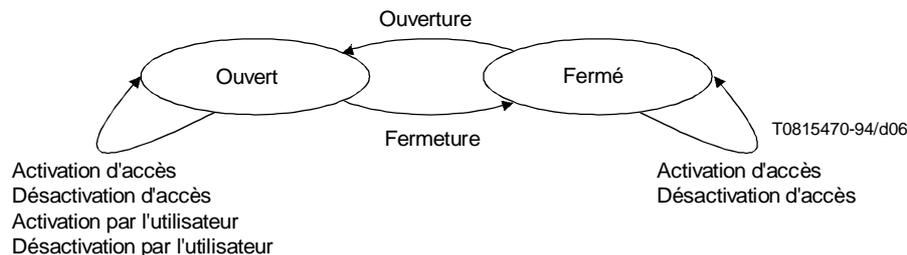


FIGURE 6/T.107

Diagramme d'état pour les composants

En général, les états logiques suivants sont définis pour les composants, avec certaines restrictions en ce qui concerne des objets spécifiques (voir l'article 7):

- Etat *ouvert*:

dans cet état, le composant qui existe dans un objet est affiché.

L'ouverture d'un composant est lancée par l'application VEMMI, soit en tant qu'état initial du composant au moment de la création de l'objet, par une commande VEMMI_Modify_Component à un moment quelconque de l'application ou par une action locale associée à un composant et déclenchée par l'interaction avec l'utilisateur.

- Etat *fermé*:

dans cet état, le composant qui existe dans le terminal n'est pas affiché.

La fermeture d'un composant est lancée par l'application VEMMI, soit en tant qu'état initial du composant au moment de la création de l'objet, par une commande VEMMI_Modify_Component à un moment quelconque de l'application ou par une action locale associée à un composant et déclenchée par l'interaction avec l'utilisateur.

4.7.2.2 Définition des paramètres d'état des composants

En général, les paramètres d'état suivants sont définis pour les composants, avec certaines restrictions en ce qui concerne des composants spécifiques (voir l'article 7):

- Etat *actif*:
dans cet état, un composant est ce vers quoi la saisie converge actuellement. Les saisies d'utilisateur se rapportent toujours au composant actif situé dans l'objet actif. Si le composant actif est inaccessible, l'utilisateur ne peut pas entrer en interaction avec lui.
La gestion de l'état actif est décrite au 4.10.1.
- Etat *inactif*:
état inverse de l'état actif.
- Etat *accessible*:
l'utilisateur peut entrer en interaction avec un composant à l'état accessible, si ce composant fait partie de l'objet actif actuel.
C'est l'application VEMMI qui permet à l'utilisateur d'entrer en interaction avec un composant, soit en tant qu'état initial au moment de la création de l'objet, par une commande VEMMI_Modify_Component à n'importe quel moment de l'application ou par une action locale associée à un composant et déclenchée par l'interaction avec l'utilisateur.
- Etat *inaccessible*:
état inverse de l'état accessible.

4.8 Gestion d'action locale

Les commandes VEMMI sont généralement émises par l'application VEMMI afin de modifier l'état ou les paramètres d'état d'un objet ou d'un composant. Seul un nombre très limité de changements d'état ou de paramètre d'état d'objet ou de composant peut être lancé par le terminal.

Pour que l'application VEMMI puisse entrer en interaction avec l'utilisateur, le terminal doit offrir la capacité de rendre compte des données saisies par l'utilisateur sur l'accès ou la validation des composants.

Le résultat de ces interactions est que des changements d'état d'objet ou de composant, dus à des saisies par l'utilisateur, subiront un retard. Les saisies effectuées par l'utilisateur devront en effet être envoyées à l'application VEMMI avant que celle-ci puisse réagir par une commande VEMMI de modification d'état. Cette procédure peut entraîner des temps de réponse non acceptables sur des liaisons de données à faible débit. Un ensemble d'actions locales a été défini pour améliorer les temps de réponse.

Les actions locales font partie des définitions des composants. Elles sont mémorisées dans le terminal au moment de la création de l'objet. Chaque action locale peut se composer d'une liste de «commandes élémentaires spécifiques», d'une liste de «commandes de compte rendu» et d'une liste de «commandes générales» dans un ordre quelconque. On distingue les commandes suivantes:

- a) *Commandes élémentaires spécifiques*:
 - ouverture de composants de l'objet de rattachement (paramètre: liste de CIN);
 - fermeture de composants de l'objet de rattachement (paramètre: liste de CIN);
 - ouverture d'objets (paramètre: liste de OIN);
 - fermeture d'objets (paramètre: liste de OIN);
 - modification d'un paramètre d'état de composant situé dans l'objet de rattachement, pour le faire passer à l'état inaccessible (paramètre: liste de CIN);
 - modification d'un paramètre d'état de composant situé dans l'objet de rattachement, pour le faire passer à l'état accessible (paramètre: liste de CIN);
 - effacement d'objets (paramètre = liste de OIN);
 - ouverture de blocage d'objet (paramètre = OIN).
- b) *Commandes de compte rendu*:
 - signalisation des OIN et CIN;
 - signalisation des valeurs actuelles du composant;

- signalisation des valeurs de tous les composants de l'objet de rattachement;
 - signalisation des valeurs des composants de l'objet de rattachement qui ont été modifiées (également par l'intermédiaire d'une commande VEMMI_Modify_Component) depuis le dernier compte rendu ou depuis la création de l'objet s'il s'agit du premier compte rendu.
- c) *Commandes générales:*
- blocage d'utilisateur;
 - réinitialisation des valeurs de tous les composants introduits dans l'objet de rattachement. Cette commande ne doit être appliquée que si l'attribut «StoreInitialValue» de l'objet créé a pour valeur true.

Ces commandes font partie de la définition des objets et peuvent apparaître dans l'action locale d'un composant dans n'importe quel ordre. Elles doivent être exécutées dans l'ordre où elles apparaissent. Il existe deux événements spécifiques de déclenchement d'actions locales:

- activation d'un composant;
- validation d'un composant.

On peut utiliser une liste de «commandes élémentaires spécifiques», une liste de «commandes de compte rendu» et une liste de «commandes générales» pour spécifier une même action locale relative à un composant. Aucune commande d'un type particulier n'est obligatoire pour une action locale.

Il existe deux événements spécifiques de déclenchement d'actions locales:

- activation d'un composant;
- validation d'un composant.

Une action locale peut être associée à chacun de ces événements de déclenchement.

4.9 Considérations relatives à la mise en mémoire

Les terminaux VEMMI sont censés avoir des capacités de mémorisation correctement calculées afin d'assurer un service VEMMI satisfaisant, en étroite rapport avec leur mode de traitement des contenus de type de données.

Lorsque l'insuffisance de capacité de la mémoire rend impossible l'exécution d'une commande VEMMI ou d'une action locale, le gestionnaire local peut décider d'effacer certains objets à l'état fermé. Ce mécanisme dépend du terminal utilisé.

Le terminal VEMMI doit envoyer un message VEMMI_Error (mémoire insuffisante) à l'application VEMMI lorsqu'il n'a pas la possibilité de libérer assez de mémoire pour traiter une description d'objet reçue avec une commande VEMMI_Create_Object, une commande VEMMI_Modify_Component ou une commande VEMMI_Open_Object. Dans ce cas, le terminal peut ne pas tenir compte de commandes de type VEMMI_Create_Object ultérieures, tant que l'application VEMMI n'a pas libéré une tranche suffisante de mémoire par l'intermédiaire d'une commande de type VEMMI_Close_Object ou VEMMI_Destroy_Object. L'application VEMMI peut également décider de déconnecter la ligne si l'application ne peut pas être exécutée correctement en raison du manque de mémoire dans le terminal; l'utilisateur doit cependant toujours être informé de cette décision, par un message déposé dans une boîte de messages.

4.10 Règles communes de traitement des objets

4.10.1 Gestion d'état d'activité et de convergence

Un objet VEMMI peut passer à l'état actif consécutivement aux événements suivants:

- a) ouverture par l'application VEMMI;
- b) accès par l'utilisateur;
- c) action locale déclenchant la commande élémentaire spécifique «open object»;
- d) fermeture de l'objet à l'état actif. L'objet ouvert qui était le plus récemment à l'état actif devient le nouvel objet actif.

L'état d'activité doit être conféré au dernier objet VEMMI ouvert, jusqu'à ce que l'utilisateur accède à un autre objet VEMMI ouvert. Lorsqu'un objet VEMMI ouvert et accessible est consulté par l'utilisateur, le gestionnaire local doit conférer à cet objet l'état actif. Cet état sera ensuite traité par le gestionnaire local jusqu'à ce que l'application VEMMI lance une action d'ouverture ou de réouverture d'un autre objet VEMMI. La DDA doit toujours contenir un seul objet actif. Celui-ci peut être inaccessible.

Si des objets font appel à une aire commune de la DDA, les activations successives de ces objets peuvent provoquer des superpositions partielles ou totales d'objets. L'objet actif doit toujours être visible en totalité par l'utilisateur, dès qu'il a été activé. L'objet actif précédent est logiquement placé immédiatement sous l'objet actif. Cette règle est applicable à tous les objets déjà activés.

Lorsqu'il est mis dans l'état actif, l'objet VEMMI et/ou un de ses composants doivent recevoir du gestionnaire local l'indication de leur point de convergence. L'état d'activité de l'objet VEMMI et/ou de ses composants VEMMI doit être affiché de manière claire sur l'écran de l'utilisateur. Dans l'objet VEMMI actif, un seul composant VEMMI accessible doit recevoir l'état d'activité et l'indication du point de convergence.

La première fois qu'un objet est affiché, et par conséquent activé, le composant auquel le point de convergence est indiqué peut être spécifié par l'application VEMMI dans la description de l'objet. Si aucun composant n'a été spécifié ou si le composant spécifié n'est pas ouvert, le terminal doit indiquer le point de convergence au premier composant ouvert et accessible créé. Dans les autres cas de figure, le mode d'indication du point de convergence au moment de l'activation d'un objet à un de ses composants dépend du terminal utilisé (même règle que ci-dessus, mémorisation du dernier composant consulté, le plus proche de l'objet de rattachement consulté par l'utilisateur).

4.10.2 Comportement du mode modal

Pour un objet modal actif, l'entrée en interaction de l'utilisateur avec un autre objet VEMMI de la même application VEMMI est interdite par le terminal, l'utilisateur étant prévenu de toute tentative en ce sens de sa part (avertisseur sonore).

Toutes les commandes communiquées par le serveur et toutes les actions locales consécutives aux différentes fois où l'utilisateur est entré en interaction avec les composants de l'objet modal sont exécutées.

4.10.3 Considérations relatives aux dimensions et au découpage

Le contenu de données d'un élément VEMMI doit être expédié avec des dimensions et une éventuelle position compatibles avec la taille décrite pour son conteneur (objet ou composant). Un terminal VEMMI doit découper un contenu de données non compatibles pour l'ajuster aux dimensions globales du conteneur.

Des règles propres au contenu de données en phototrame sont applicables; elles sont indiquées avec la description des éléments.

4.11 Mémorisation d'objets locaux

Une application assurée par le serveur peut demander l'introduction d'objets dans la mémoire locale du terminal. Un tel ensemble d'objets est propre à l'application et contient normalement les parties qui restent inchangées plus durablement et qu'il est par conséquent préférable, pour plus d'efficacité, de ne pas transmettre au terminal dans chaque session de dialogue. Les objets ainsi introduits sont traités de la même manière que s'ils avaient été reçus pendant la session actuelle. A la demande du serveur, ils peuvent être restitués aux fins de leur mise à jour. Il n'est pas nécessaire d'extraire les objets de l'application du terminal pour les mémoriser; on se contente d'en mémoriser une copie. Le dialogue peut se poursuivre sans être affecté par la mémorisation des objets. On identifie en ensemble d'objets à l'aide de l'identificateur d'application.

L'indication de la date et de l'heure (timestamp, horodatage) est donnée par le serveur au début de l'application. Elle est mémorisée avec les objets et réutilisée au cours de sessions ultérieures. La valeur d'horodatage n'est jamais calculée par le terminal, n'étant donnée qu'à titre de comparaison pour déceler les objets périmés. Chaque fois qu'un objet est mémorisé localement dans le terminal, une indication d'horodatage doit lui être associée. La valeur de cette indication doit être déterminée comme suit:

- si l'objet a été créé par le serveur (au moyen de la commande VEMMI_Create_Object) pendant la session actuelle ou si un quelconque de ses composants a été modifié (au moyen de la commande VEMMI_Modify_Component) pendant la session actuelle, il doit être mémorisé avec la valeur d'horodatage indiquée par la commande VEMMI_Open_Application pour l'application actuelle;
- dans tous les autres cas, l'indication d'horodatage des objets doit rester inchangée (l'objet introduit a été extrait tel quel de la mémoire locale pendant la présente session).

En outre, deux attributs, qui font partie de la définition de l'objet, permettent de demander que celui-ci soit mémorisé, immédiatement après sa création (gabarit) ou à la fin de l'application (autostore, mémorisation automatique).

Un ensemble d'objets mémorisés contient les éléments suivants:

- l'identificateur d'application;
- les objets assortis de leurs composants et de l'indication d'horodatage correspondante;

- les fichiers désignés (si le terminal ne peut accéder à l'un quelconque de ces fichiers, l'objet est considéré comme étant incomplet et il ne doit pas être mémorisé);
- la table de couleurs appropriée introduite.

Le format de mémorisation est propre au terminal utilisé, les objets ne pouvant être introduits et mémorisés que par ce même terminal.

Le terminal doit être d'un modèle qui permette à l'utilisateur de gérer de manière appropriée la mémorisation locale et, en règle générale, de limiter la taille maximale de la mémoire.

4.12 Noms d'annuaire symboliques

Afin d'aider l'application assurée par le serveur à localiser les bons fichiers, il est défini des noms d'annuaire symboliques qui peuvent constituer une partie du paramètre «filename» et que le terminal fait correspondre aux noms de fichiers d'annuaire réels. Dans l'exemple qui suit, une application assurée par le serveur utilise un nom d'annuaire symbolique pour identifier un fichier en phototrame:

VMMI_Create_Object (OIN = 7, ..., nom de fichier = «\$CD\ARTISTS\DANCER\LISA.BMP», type de fichier d'image = BMP).

La chaîne «\$CD» identifie l'unité CD-ROM. Le terminal remplace ce nom symbolique par l'identification réelle de l'unité. L'identification complète du fichier peut être la suivante: «H:\ARTISTS\DANCER\LISA.BMP».

Les noms symboliques ci-dessous sont définis; l'application peut utiliser d'autres noms symboliques:

Nom	Signification
\$APL	Annuaire de rattachement des applications mémorisées dans le terminal
\$FTD	Annuaire de rattachement des fichiers transférés
\$CD	Nom d'annuaire d'une unité CD-ROM

Ces noms symboliques peuvent être utilisés dans le paramètre «filename» (nom de fichier) des commandes VEMMI. Le nom symbolique ne doit figurer qu'une seule fois dans le paramètre nom de fichier. Pour séparer le nom symbolique des sous-répertoires supplémentaires et du nom de fichier, il convient d'utiliser le caractère «\», code 5/12.

4.13 Règles propres aux terminaux spécialisés

Toutes les primitives de service et leurs paramètres doivent être compris également par un terminal spécialisé. Pour l'exécution des commandes liées à des fonctions particulières, un terminal spécialisé peut appliquer des mécanismes de dégradation progressive.

Un terminal spécialisé peut assurer la dégradation progressive des fonctions suivantes:

- définition des couleurs;
- attributs in-text (polices, couleurs, attributs de police);
- mémorisation locale;
- positionnement et dimensionnement des NDC.

Sur les terminaux VEMMI spécialisés n'ayant que la capacité de traiter un contenu de données vidéotex en mode caractère (alphamosaïques, redéfinissables, du profil photographique 1), les objets et composants VEMMI peuvent être physiquement positionnés et dimensionnés d'après la position de caractère affichable correspondante la plus proche. Afin d'éviter des effets secondaires entre objets VEMMI, il convient donc d'exprimer les NDC de position des objets en multiples directs des positions de caractères dans le cas d'applications VEMMI n'utilisant que ce type de contenu de données vidéotex.

Sur les terminaux VEMMI spécialisés n'ayant que la capacité de traiter un contenu de données vidéotex en mode caractère, on peut utiliser VEMMI avec le mode 80 caractères en utilisant l'élément de données de présentation vidéotex (VPDE) approprié SWITCHING SEQUENCE ou DEFINE FORMAT pour passer à la syntaxe de données souhaitée. Le mode 80 caractères n'est pas obligatoire pour un terminal VEMMI.

Les terminaux VEMMI spécialisés doivent reconnaître, sinon nécessairement traiter, la totalité ou une partie des attributs en mode texte de haute qualité VEMMI qu'ils ne sont pas en mesure d'afficher. Ces terminaux peuvent ne pas tenir compte des attributs en mode texte qu'ils ne sont pas en mesure de traiter et afficher le texte à l'aide de la police du système, sur la zone de fond (arrière plan) de celui-ci, dans la couleur disponible.

Les terminaux VEMMI spécialisés doivent admettre les différents types d'objets VEMMI, sous réserve d'éventuelles restrictions s'appliquant aux objets opératifs. Ces terminaux ne devant pas obligatoirement être munis d'un système de gestion de fichier, ils peuvent gérer les fichiers qu'ils reçoivent au moyen de la commande VEMMI_Resource_Transfer dans leur propre représentation interne.

Le terminal spécialisé doit accepter le noyau minimal de types de données défini dans l'Annexe C. Toutefois, lorsque le terminal est utilisé dans un service de consultation où le mode standard est le mode vidéotex et qu'il n'accepte pas la syntaxe photographique vidéotex (ce que TFI permettra de savoir) la prise en charge des formats JPEG et GIF n'est pas obligatoire. Lorsque le terminal accepte la syntaxe photographique vidéotex, la prise en charge du format JPEG est obligatoire; la prise en charge du format GIF est recommandée. Dans les mêmes conditions de service, lorsqu'un terminal n'accepte pas la syntaxe audio vidéotex, la prise en charge des formats WAVE et MIDI n'est pas obligatoire. Lorsque le terminal accepte la syntaxe audio vidéotex, la prise en charge du format WAVE est obligatoire; la prise en charge du format MIDI est recommandée.

Les terminaux spécialisés ne doivent pas obligatoirement accepter la présentation à géométrie variable de données en phototrame permettant de remplir la zone d'affichage élémentaire. Si l'attribut de présentation d'une phototrame est mis sur «géométrie variable», le terminal peut présenter cette phototrame en juxtaposé.

5 Description du service

Cet article décrit les caractéristiques offertes par le service VEMMI.

Ce service est défini entre un terminal et une application distante. Aucune hypothèse n'est formulée quant à la façon dont le terminal accède au service [réseau téléphonique public commuté (RTPC)/réseau public pour données à commutation par paquets (RPDCP)/réseau public avec intégration des services (RNIS)] ni à celle dont la connexion est établie.

Tous les éléments du service VEMMI, ou toutes les commandes VEMMI, ne doivent être émis que sur une connexion établie dans un réseau. Avant d'envoyer la première commande VEMMI, l'application VEMMI doit normalement émettre une demande d'identificateur de profil de terminal (TFI) afin de faire en sorte que le terminal puisse accepter la norme VEMMI.

Tous les éléments du service VEMMI sont obligatoires pour un terminal VEMMI. Un tel terminal doit comprendre tous les éléments de service VEMMI, y compris tous leurs paramètres.

Les services VEMMI se subdivisent en deux groupes:

- les services lancés par l'application VEMMI;
- les services lancés par le terminal VEMMI.

Une deuxième source de commandes pour un terminal VEMMI est l'objet de métacode. Voir le Tableau 2.

Les services suivants sont expressément confirmés:

- l'élément VEMMI_Identify_Term_Cap est confirmé par l'élément VEMMI_Identify_Term_Cap_Resp;
- l'élément VEMMI_Open_Application est confirmé par l'élément VEMMI_Open_Application_Resp;
- l'élément VEMMI_Store_Objects est confirmé par l'élément VEMMI_Store_Objects_Resp.

Aucun des autres éléments de service n'est expressément confirmé.

Les services VEMMI, à l'exception du service VEMMI_Open_Object, qui se rapportent à un objet inexistant, ne doivent pas être pris en considération par le terminal VEMMI. Si un service VEMMI_Open_Object se rapporte à un objet inexistant, le terminal doit demander que cet objet soit retransmis.

NOTE – Dans le paragraphe qui suit, tous les services VEMMI sont décrits mais seuls les paramètres nécessaires à la compréhension du service sont indiqués. Pour la liste complète des paramètres, voir l'article 8.

TABLEAU 2/T.107

Services VEMMI

Services VEMMI	Lancé par Application/Terminal	Fonction
VEMMI_Set_Translation_Mode	Application/Terminal	Sélection du mode de conversion des données VEMMI
VEMMI_Open	Application	Commutation sur mode VEMMI et réinitialisation
VEMMI_Close	Application	Commutation sur mode standard et réinitialisation
VEMMI_Suspend	Application	Commutation sur mode standard
VEMMI_Resume	Application	Commutation sur mode VEMMI
VEMMI_Identify_Term_Cap	Application	Demande d'identification des capacités du terminal
VEMMI_Identify_Term_Cap_Resp	Terminal	Réponse à la demande d'identification des capacités du terminal»
VEMMI_Set_Options	Application	Sélection d'options dans le terminal
VEMMI_Create_Object	Application	Définition d'objet
VEMMI_Open_Object	Application	Affichage d'objet
VEMMI_Open_Blocking_Object	Application	Ouverture d'objet
VEMMI_Close_Object	Application	Effacement d'un objet de l'écran
VEMMI_Close_All	Application	Effacement de tous les objets sur l'écran
VEMMI_Destroy_Object	Application	Effacement d'objet dans la mémoire du terminal
VEMMI_Obj_Access_Disable	Application	Interdiction d'accès utilisateur
VEMMI_Obj_Access_Enable	Application	Autorisation d'accès utilisateur
VEMMI_Modify_Component	Application	Modification de caractéristiques de composant
VEMMI_Obj_Location_Change	Application	Définition d'un nouvel emplacement d'objet
VEMMI_Load_Col_Table	Application	Introduction d'une table de couleurs
VEMMI_Reset_Col_Table	Application	Remise à l'état initial d'une table de couleurs
VEMMI_Open_Application	Application	Ouverture d'une application
VEMMI_Open_Application_Resp	Terminal	Réponse à une demande d'«ouverture d'application»
VEMMI_Delete_Outdated_Objects	Application	Effacement des objets périmés
VEMMI_Store_Objects	Application	Mémorisation d'un ensemble d'objets
VEMMI_Store_Objects_Resp	Terminal	Réponse à une demande de «mémorisation d'objets»
VEMMI_Erase_Objects	Application	Suppression d'objets de la mémoire locale
VEMMI_User_Lock	Application	Interdiction de saisies utilisateur
VEMMI_User_Unlock	Application	Autorisation de saisies utilisateur
VEMMI_Object_Retransmission	Terminal	Demande de retransmission d'un objet
VEMMI_User_Data	Terminal	Données d'utilisateur en mode VEMMI
VEMMI_Error	Terminal	Condition ou situation d'erreur
VEMMI_Resource_Transfer	Application	Transfert de ressources
VEMMI_Resource_Transfer_Abort	Terminal	Abandon-transfert de ressources

5.1 Eléments de service lancés par l'application et le terminal VEMMI

Lorsqu'il est émis par l'application VEMMI, cet élément de service est utilisé pour indiquer que les données VEMMI envoyées au terminal par l'application sont transcodées selon un mode de conversion spécifié.

S'il est émis par le terminal VEMMI, cet élément de service est utilisé pour indiquer que les données VEMMI envoyées par le terminal à l'application sont transcodées selon un mode de conversion spécifié.

Etant donné qu'un terminal VEMMI commence généralement à fonctionner, par défaut, dans un environnement à 7 bits ou à 8 bits, une application VEMMI n'aura pas obligatoirement à envoyer comme première commande VEMMI dans une session VEMMI une commande VEMMI_Set_Translation_Mode pour faire passer le terminal à un mode de conversion souhaité. Dans le cas où l'application ignore si tous les terminaux en communication fonctionnent dans un environnement déterminé, elle doit envoyer une commande de service VEMMI_Set_Translation_Mode pour permettre au terminal d'interpréter correctement les commandes VEMMI. Voir le Tableau 3.

En règle générale, la commande de service VEMMI_Set_Translation_Mode est la première commande VEMMI d'une session VEMMI, mais elle peut être utilisée à tout moment au cours d'une session VEMMI.

NOTE – Le mode de conversion peut ne pas être le même dans les deux sens.

TABLEAU 3/T.107

Paramètres

Paramètres	Description
TranslationMode	Un octet spécifiant le mode de conversion souhaité

Le paramètre TranslationMode peut prendre les valeurs suivantes:

- 2/0: pas de conversion (transparent);
- 2/1: codage 3 sur 4 (voir la Recommandation T.101 [4]);
- 2/2: codage par décalage en 7 temps (voir la Recommandation T.101 [4]).

5.2 Eléments de service lancés par l'application VEMMI

5.2.1 VEMMI_Open (ouverture VEMMI)

Cet élément de service doit être utilisé pour commuter le terminal VEMMI sur le mode de fonctionnement VEMMI. Le mécanisme de commutation est défini au 4.4.3. En mode de fonctionnement VEMMI, le mécanisme décrit au 4.4.2 est applicable.

Outre la commutation sur le mode de fonctionnement VEMMI, cet élément de service doit assurer les opérations de réinitialisation suivantes:

- fermeture d'une application ouverte;
- effacement de tous les objets de l'ensemble d'objets se trouvant à l'état «ouvert» ou à l'état «fermé»;
- effacement de tous les objets ouverts sur l'écran;
- libération de tous les numéros d'identification des objets ressources;
- réglage de la table de couleurs aux valeurs prédéfinies.

Les objets mémorisés dans une mémoire locale ne sont pas altérés.

Si cette commande est reçue en mode VEMMI, aucune commutation n'est nécessaire; seules les opérations de réinitialisation sont exécutées. Voir le Tableau 4.

TABLEAU 4/T.107

Paramètres

Paramètres	Description
Version	Identificateur de la version VEMMI
PrivateMode	Paramètre privé qui peut être utilisé dans certains services vidéotex «par page». Ce mode n'est pas décrit dans la présente Recommandation

5.2.2 VEMMI_Close (fermeture VEMMI)

Cet élément de service doit être utilisé pour commuter le terminal VEMMI sur le mode standard. Le mécanisme de commutation est défini au 4.4.3. En mode de fonctionnement standard, le mécanisme décrit au 4.4.1 est applicable.

Outre la commutation sur le mode standard, cet élément de service doit exécuter les opérations de réinitialisation définies au 5.2.1. Voir le Tableau 5.

TABLEAU 5/T.107

Paramètres

Paramètres	Description
Aucun	

5.2.3 VEMMI_Resume (reprise VEMMI)

Cet élément de service doit être utilisé pour commuter le terminal VEMMI sur le mode de fonctionnement VEMMI. Les paramètres de cet élément de service sont les mêmes que ceux de l'élément de service VEMMI_Open (voir 5.2.1). Cet élément de service ne nécessite l'exécution d'aucune opération de réinitialisation. Voir le Tableau 6.

TABLEAU 6/T.107

Paramètres

Paramètres	Description
Version	Identificateur de la version VEMMI
PrivateMode	Paramètre privé qui peut être utilisé dans certains services vidéotex «par page». Ce mode n'est pas décrit dans la présente Recommandation

5.2.4 VEMMI_Suspend (suspension VEMMI)

Cet élément de service doit être utilisé pour commuter le terminal VEMMI sur le mode standard. Aucune opération de réinitialisation ne doit être exécutée. Voir le Tableau 7.

TABLEAU 7/T.107

Paramètres

Paramètres	Description
Aucun	

5.2.5 VEMMI_Identify_Term_Cap (identification des capacités du terminal VEMMI)

Cet élément de service doit être utilisé pour demander l'identification des capacités du terminal. Voir le Tableau 8.

Les informations suivantes peuvent être obtenues:

- identification de la version VEMMI;
- mémorisation locale (acceptée/non acceptée);
- liste des langues préférées de l'utilisateur (chaîne);
- type de système (par exemple: IBM compatible, Macintosh, ...), système d'exploitation (par exemple: MS-DOS, OS2, Microsoft Windows, ...), version de système d'exploitation (par exemple: 3.1, ...) (chaîne);
- liste des types de données acceptés (rappel du catalogue d'identification de codage de contenu VEMMI).

Si le paramètre DesiredInformation n'est pas présent, le terminal doit répondre en déclinant la liste complète de ses capacités.

TABLEAU 8/T.107

Paramètres

Paramètres	Description
DesiredInformation	Type d'information demandé au terminal

5.2.6 VEMMI_Set_Options (sélection d'options VEMMI)

Cet élément de service doit être utilisé pour choisir les options du terminal. Voir le Tableau 9.

Les options suivantes peuvent être choisies:

- si le terminal accepte différentes formes de codage de texte, la norme de codage souhaitée peut être sélectionnée.

TABLEAU 9/T.107

Paramètres

Paramètres	Description
TextStandard	Rappel d'une norme de codage de texte dans le catalogue d'identification de codage de contenu VEMMI, à utiliser au début du codage de la chaîne et des caractères des types de données

5.2.7 VEMMI_Create_Object (création d'objet VEMMI)

Cet élément de service est utilisé pour définir un objet VEMMI et ses composants dans le terminal VEMMI. Celui-ci doit créer l'objet et appliquer l'état d'objet et les paramètres d'état d'objet qui sont indiqués dans la définition de cet objet.

Les OIN doivent être uniques dans le temps pour une application donnée. La création d'un objet avec un OIN déjà attribué doit provoquer la destruction de l'objet précédemment numéroté.

Les CIN doivent être uniques dans le cadre d'un objet donné. La définition d'un composant avec un CIN déjà attribué doit provoquer la destruction du composant précédemment numéroté. Voir le Tableau 10.

TABLEAU 10/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	
Gabarit (Template)	Si une application est ouverte, l'objet doit être mémorisé immédiatement après sa création
Mémorisation automatique (Autostore)	Si une application est ouverte, l'objet doit être mémorisé à la réception d'une commande VEMMI_Open, VEMMI_Close ou VEMMI_Open_Application

5.2.8 VEMMI_Open_Object (ouverture d'objet VEMMI)

Cet élément de service est utilisé pour mettre un objet dans l'état ouvert dans le terminal VEMMI. Ce terminal doit afficher immédiatement l'objet spécifié. Si cet objet n'est pas présent dans le terminal, celui-ci doit demander cet objet en envoyant à l'application VEMMI une demande VEMMI_Object_Retransmission avec l'OIN spécifié, auquel cas la «règle A» exposée au 5.3.2 doit être appliquée. L'application doit alors recréer l'objet, en lui appliquant son état actuel et ses paramètres d'état actuels.

Une commande VEMMI_Open_Object se rapportant à un objet déjà ouvert provoque l'activation de cet objet.

Si une commande VEMMI_Open_Object est appliquée à un objet ressource, le terminal doit vérifier si l'objet est disponible. Si l'objet n'est pas disponible, le terminal doit le demander. Aucune autre action ne doit être exécutée.

Si une commande VEMMI_Open_Object est appliquée à un objet de métacode, le terminal exécute le contenu de l'objet. Si cet objet n'est pas présent, il faut le demander.

Si une commande VEMMI_Open_Object est appliquée à un objet opératif, le terminal exécute le programme. Si cet objet n'est pas présent, il faut le demander. Voir le Tableau 11.

TABLEAU 11/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	

5.2.9 VEMMI_Open_Blocking_Object (ouverture d'objet VEMMI avec blocage)

Cet élément de service assure la même fonction que l'élément de service VEMMI_Open_Object, à ceci près que la règle de retransmission applicable est cette fois la «règle B», décrite au 5.3.2.

5.2.10 VEMMI_Close_Object (fermeture d'objet VEMMI)

Cet élément de service est utilisé pour fermer un objet VEMMI dans le terminal VEMMI. Celui-ci doit effacer de l'écran l'objet indiqué, mais le garder en mémoire. Les valeurs et paramètres d'état actuels des composants ne doivent pas être modifiés.

Lorsque le terminal reçoit une primitive d'indication VEMMI_Close_Object se rapportant à un objet actif, le nouvel objet actif affiché sur ce terminal doit être l'objet ouvert qui était le plus récemment à l'état actif dans le terminal.

Pour les objets non affichables ou qui sont déjà à l'état fermé, la commande VEMMI_Close_Object est sans effet. Voir le Tableau 12.

TABLEAU 12/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	

5.2.11 VEMMI_Close_All (fermeture de tous les objets VEMMI)

Cet élément de service doit être utilisé pour fermer tous les objets VEMMI dans le terminal VEMMI. Un élément de service VEMMI_Close_All ne doit pas modifier le mode de fonctionnement actuel du terminal. Voir le Tableau 13.

TABLEAU 13/T.107

Paramètres

Paramètres	Description
Aucun	

5.2.12 VEMMI_Destroy_Object (effacement d'objet VEMMI)

Cet élément de service doit être utilisé pour supprimer un objet VEMMI de l'ensemble d'objets qui sont à l'état «ouvert» ou «fermé». Celui-ci doit effacer de sa mémoire toutes les données se rapportant à cet objet. Si le service VEMMI_Destroy_Object est appliqué à un objet à l'état ouvert, cet objet doit être fermé par le gestionnaire local puis effacé. Si cet objet était à l'état actif, le nouvel objet actif dans le terminal VEMMI doit être l'objet ouvert qui était le plus récemment à l'état actif dans le terminal. Voir le Tableau 14.

TABLEAU 14/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	

5.2.13 VEMMI_Obj_Access_Disable (interdiction d'accès à l'objet VEMMI)

Cet élément de service est utilisé pour empêcher l'utilisateur d'avoir accès à un objet dans le terminal VEMMI. L'accès doit rester interdit jusqu'à ce qu'un élément de service VEMMI_Obj_Access_Enable, se rapportant au même objet, soit reçu par le terminal VEMMI.

Si cet objet était à l'état actif, le nouvel objet actif doit être l'objet ouvert qui était le plus récemment à l'état actif dans le terminal. Une primitive d'indication VEMMI_Obj_Access_Disable se rapportant à un objet actif doit interrompre l'interaction de l'utilisateur avec cet objet. Voir le Tableau 15.

TABLEAU 15/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	

5.2.14 VEMMI_Obj_Access_Enable (autorisation d'accès à l'objet VEMMI)

Cet élément de service doit être utilisé pour autoriser l'utilisateur à avoir accès à un objet dans le terminal VEMMI. L'accès doit être autorisé jusqu'à ce qu'un élément de service VEMMI_Obj_Access_Disable, se rapportant au même objet, soit reçu par le terminal VEMMI.

Une primitive d'indication VEMMI_Obj_Access reçue dans le terminal VEMMI et se rapportant à un objet inactif ne doit pas modifier ce paramètre d'état. Voir le Tableau 16.

TABLEAU 16/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	

5.2.15 VEMMI_Modify_Component (modification de composant VEMMI)

Cet élément de service sert à modifier un ou plusieurs composant(s) VEMMI appartenant à un objet VEMMI créé dans le terminal VEMMI. Les modifications qui peuvent être apportées à des composants VEMMI sont définies dans les tableaux appropriés du 9.6.

Si une commande VEMMI_Modify_Component est appliquée à un objet à l'état ouvert, cela doit apparaître immédiatement sur l'écran du terminal VEMMI si la partie modifiée est visible.

Une commande VEMMI_Modify_Component ne doit pas modifier le paramètre d'état actif/inactif de l'objet se trouvant dans le terminal VEMMI. Voir le Tableau 17.

TABLEAU 17/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	
Description de modification	

5.2.16 VEMMI_Obj_Location_Change (modification d'emplacement d'objet VEMMI)

Cet élément de service ne sert qu'à modifier l'emplacement d'un objet VEMMI (et non d'un composant) sur l'écran d'un terminal VEMMI. Si une commande VEMMI_Obj_Location_Change se rapporte à un objet VEMMI à l'état fermé, aucun effet visible ne doit être constaté. Cet élément de service ne doit pas modifier le paramètre d'état actif/inactif d'un objet.

Un élément de service VEMMI_Obj_Location_Change ne doit jamais être appliqué à une barre d'application. Voir le Tableau 18.

TABLEAU 18/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	
XPos	Nouvelle position horizontale
YPos	Nouvelle position verticale

5.2.17 VEMMI_Load_Col_Table (introduction de la table de couleurs VEMMI)

Cet élément de service doit être utilisé par le serveur pour définir les couleurs de la table de couleurs.

Le serveur définit les couleurs l'une après l'autre en commençant par l'indice «ColEntry». Le terminal doit envoyer un message d'erreur au serveur, si celui-ci ne peut définir toutes les couleurs qu'il a demandées du fait, par exemple, que l'équipement ne les accepte pas toutes. Voir le Tableau 19.

TABLEAU 19/T.107

Paramètres

Paramètres	Description
ColEntry	Indice de couleur d'après lequel les couleurs sont définies par ordre croissant
ColRGBList	Liste de triplets indiquant les composantes chromatiques rouge, vert et bleu d'une couleur. La valeur de chacune de ces composantes chromatiques se situe dans la gamme 0 .. 63. La valeur 0 correspond à l'absence totale d'intensité de la composante chromatique, la valeur 63 correspondant à son intensité maximale

NOTE – Si ce service n'est pas pris en charge par le terminal, on doit le signaler à l'application en cours en lui envoyant le message VEMMI_Error «service non pris en charge».

5.2.18 VEMMI_Reset_Col_Table (remise à l'état initial de la table de couleurs VEMMI)

Cet élément de service doit être utilisé par le serveur pour remettre à l'état initial toutes les couleurs précédemment définies. Cette commande ne comporte aucun paramètre. Seules les couleurs par défaut (indices 0 .. 15) restent définies. Si les couleurs par défaut ont été redéfinies au moyen de la commande VEMMI_Load_Col_Table, la commande VEMMI_Reset_Col_Table donne aux couleurs par défaut leurs valeurs prédéfinies. Voir le Tableau 20.

TABLEAU 20/T.107

Paramètres

Paramètres	Description
Aucun	

Les couleurs par défaut sont indiquées dans le Tableau 21.

TABLEAU 21/T.107

Paramètres

Indice de couleur	Rouge	Vert	Bleu	Couleur
0	0	0	0	Noir
1	0	0	42	Bleu
2	0	42	0	Vert
3	0	42	42	Cyan
4	42	0	0	Rouge
5	42	0	42	Magenta
6	42	21	0	Brun
7	42	42	42	Blanc
8	21	21	21	Gris
9	21	21	63	Bleu clair
10	21	63	21	Vert clair
11	21	63	63	Cyan clair
12	63	21	21	Rouge clair
13	63	21	63	Magenta clair
14	63	63	21	Jaune
15	63	63	63	Blanc éclatant

NOTE – Si ce service n'est pas pris en charge par le terminal, on doit le signaler à l'application en cours en lui envoyant le message VEMMI_Error «service non pris en charge».

5.2.19 VEMMI_Open_Application (ouverture d'application VEMMI)

Cet élément de service doit être utilisé par le serveur pour introduire un ensemble d'objets et pour définir le nom et les attributs de l'application actuelle. L'utilisation des noms d'annuaire symboliques décrits au 4.12 peut constituer une aide supplémentaire. Les ensembles d'objets mémorisés localement sont identifiés par l'identificateur d'application. Une indication d'horodatage est utilisée pour associer l'information de version aux objets de cette application. Voir le Tableau 22.

TABLEAU 22/T.107

Paramètres

Paramètres	Description
AppId	Chaîne comportant le nom de l'application
ApplAddData	Données supplémentaires d'identification d'application
Timestamp (horodatage)	Temps en secondes depuis le 1.01.1970, temps moyen de Greenwich (GMT)

Le paramètre AppId doit contenir le nom de l'application. Des informations supplémentaires, telles que le nom de la société ou de l'organisation dont relève l'application ou de l'auteur de l'application, peuvent être données au moyen du paramètre ApplAddData, qui n'est pas obligatoire. Le terminal n'a pas besoin d'analyser le contenu du paramètre AppId, qui ne sert qu'à identifier l'application demandée.

Les fournisseurs d'applications doivent prendre soin de choisir des identificateurs d'application uniques, pour éviter toute superposition d'écriture sur d'autres applications.

5.2.20 VEMMI_Delete_Outdated_Objects (effacement des objets VEMMI périmés)

Cet élément de service doit être utilisé par le serveur pour demander que les objets locaux de l'application actuelle qui sont périmés soient effacés de la mémoire du terminal. Voir le Tableau 23.

Si le serveur émet une commande VEMMI_Delete_Outdated_Objects assortie des indications d'horodatage requises pour des objets spécifiques, le terminal doit agir de la manière suivante:

- si un objet consulté a une valeur d'horodatage supérieure à la valeur d'horodatage de l'objet mémorisé localement, il doit être effacé de la mémoire locale. Le serveur est ainsi assuré, pour les besoins de son application, que les objets périmés ne sont plus utilisés.

TABLEAU 23/T.107

Paramètres

Paramètres	Description
UpdateList	Paramètre contenant des groupes de deux éléments chacun, une indication d'horodatage et un ensemble d'objets

Pour permettre la mise à jour partielle d'une application, cette commande est généralement envoyée à la suite d'une commande VEMMI_Open_Application.

5.2.21 VEMMI_Store_Objects (mémorisation d'objets VEMMI)

Cet élément de service doit être utilisé par le serveur pour demander au terminal de mémoriser les objets actuellement introduits ou une partie d'entre eux. L'indication d'horodatage de l'application actuellement ouverte, fournie avec la commande VEMMI_Open_Application, est associée à chaque objet. Voir le Tableau 24.

TABLEAU 24/T.107

Paramètres

Paramètres	Description
Numéros d'identification des objets	

Si ce paramètre est absent, tous les objets actuellement introduits sont mémorisés.

5.2.22 VEMMI_Erase_Objects (effacement d'objets VEMMI)

Cet élément de service doit être utilisé par le serveur pour demander que certains objets de l'application actuelle soient effacés de la mémoire locale d'un terminal. Pour plus de sécurité, le terminal VEMMI pourrait consulter l'utilisateur (par exemple une fois au début de l'application ou à chaque fois) avant d'effacer un objet de sa mémoire. Si aucune application n'est ouverte, la commande est sans effet. Voir le Tableau 25.

TABLEAU 25/T.107

Paramètres

Paramètres	Description
Numéros d'identification des objets	

5.2.23 VEMMI_User_Lock (blocage de l'utilisateur VEMMI)

Cet élément de service sert à interdire toute saisie par l'utilisateur dans le terminal jusqu'à ce qu'une commande VEMMI_User_Unlock soit reçue. Voir le Tableau 26.

TABLEAU 26/T.107

Paramètres

Paramètres	Description
Aucun	

5.2.24 VEMMI_User_Unlock (déblocage de l'utilisateur VEMMI)

Cet élément de service sert à autoriser les saisies par l'utilisateur dans le terminal. Voir le Tableau 27.

TABLEAU 27/T.107

Paramètres

Paramètres	Description
Aucun	

5.2.25 VEMMI_Resource_Transfer (transfert de ressources VEMMI)

Cet élément de service doit être utilisé pour transférer des fichiers désignés par des objets ressource VEMMI d'une application VEMMI à un terminal VEMMI. Voir le Tableau 28.

Un fichier peut être transféré à l'aide d'une ou de plusieurs commandes VEMMI_Resource_Transfer. Un paramètre de la commande VEMMI_Resource_Transfer (TransferID) associe la commande à un transfert de ressources donné. Cela permet d'exécuter simultanément plusieurs transferts de ressources, avec entrelacement des commandes relevant de transferts de ressources différents, sans brouillage.

Les commandes VEMMI_Resource_Transfer relevant du même transfert de ressources sont numérotées l'une après l'autre à l'aide du paramètre BlockNumber.

Le premier bloc d'un transfert de ressources donné doit toujours contenir l'information d'en-tête de fichier suivante:

- nom de fichier;
- taille de fichier;
- nombre de commandes VEMMI_Resource_Transfer utilisées pour transmettre le fichier;
- date de création.

Le paramètre TransferType de la commande VEMMI_Resource_Transfer indique si la commande VEMMI_Resource_Transfer d'un transfert de ressources donné est:

- la première commande;
- une commande intermédiaire; ou
- si le transfert de ressources est abandonné par l'application VEMMI.

TABLEAU 28/T.107

Paramètres

Paramètres	Description
TransferType	Première commande, commande intermédiaire, abandon
TransferID	Identificateur de transfert
BlockNumber	Numéro de la commande dans un transfert de ressources donné
FileInformation	Information d'en-tête concernant le fichier
TotalBlocks	Nombre de commandes VEMMI_Resource_Transfer utilisées pour transmettre un fichier donné

5.3 Eléments de service lancés par le terminal

5.3.1 VEMMI_Identify_Term_Cap_Resp (réponse à une demande d'identification des capacités du terminal VEMMI)

Cet élément de service doit être utilisé pour répondre à une demande VEMMI_Identify_Term_Cap. Voir le Tableau 29.

Les informations suivantes peuvent être données:

- identification de la version VEMMI;
- mémorisation locale (acceptée/non acceptée);
- liste des langues préférées de l'utilisateur (chaîne);
- type de système (par exemple: IBM compatible, Macintosh, ...), système d'exploitation (par exemple: MS-DOS, OS2, Microsoft Windows, ...), version de système d'exploitation (par exemple: 3.1, ...) (chaîne);
- liste des types de données acceptés (rappel du catalogue d'identification de codage de contenu VEMMI).

TABLEAU 29/T.107

Paramètres

Paramètres	Description
SupportedVEMMIVersions	Paramètre contenant l'identification de la version VEMMI
ContentList	Liste donnant les références des types de données pris en charge dans le catalogue d'identification de codage de contenu VEMMI
LocalStorage	Paramètre identifiant la prise en charge de l'option de mémorisation locale
PreferredUserLanguages	Langues préférées par l'utilisateur (pour les applications multilingues) La langue indiquée en premier sur la liste est celle que l'utilisateur préfère à toutes les autres. La liste peut ne comporter qu'une seule langue
SystemInfo	Information aidant à utiliser convenablement les objets opératifs (par exemple: type de système, système d'exploitation, version de système d'exploitation)

5.3.2 VEMMI_Object_Retransmission (retransmission d'objet VEMMI)

Cet élément de service sert à demander à l'application VEMMI de retransmettre un objet VEMMI. Il ne doit être utilisé qu'après:

- la réception d'une commande VEMMI_Open_Object;
- une commande VEMMI_Open_Blocking_Object;
- une action locale «Open object»,

se rapportant à un objet non présent dans le terminal.

Pour pouvoir poursuivre le traitement des commandes VEMMI jusqu'à ce que l'objet demandé soit reçu, le terminal doit appliquer une des règles suivantes.

Règle A

- informer l'utilisateur du délai d'attente pouvant résulter d'une retransmission (selon le terminal utilisé);
- envoyer une demande VEMMI_Object_Retransmission;
- exécuter toutes les commandes VEMMI et actions locales ne se rapportant pas à l'objet demandé;
- mémoriser toutes les commandes VEMMI et actions locales se rapportant à l'objet demandé.

Une fois l'objet demandé recréé et ouvert, le terminal doit:

- reprendre l'exécution de l'action locale éventuellement suspendue;
- reprendre l'exécution des éventuelles commandes VEMMI mémorisées reçues dans l'ordre de leur réception.

Règle B

- interdire l'accès par l'utilisateur;
- informer l'utilisateur du délai d'attente pouvant résulter d'une retransmission (selon le terminal utilisé);
- envoyer une demande VEMMI_Object_Retransmission;
- suspendre et mémoriser l'exécution des commandes VEMMI ou actions locales.

Une fois l'objet demandé recréé et ouvert, le terminal doit:

- reprendre l'exécution de l'action locale éventuellement suspendue;
- reprendre l'exécution des éventuelles commandes VEMMI mémorisées reçues dans l'ordre de leur réception;
- lever l'interdiction d'accès par l'utilisateur.

Voir le Tableau 30.

TABLEAU 30/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	

5.3.3 VEMMI_User_Data (données d'utilisateur VEMMI)

Cet élément de service doit être utilisé pour envoyer à l'application VEMMI des données d'utilisateur correspondant à un objet. Voir le Tableau 31.

TABLEAU 31/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	
CompData	Saisies par l'utilisateur

5.3.4 VEMMI_Open_Application_Resp (réponse à une commande d'ouverture d'application VEMMI)

Cet élément de service doit être utilisé par le terminal pour répondre à une commande VEMMI_Open_Application. Voir le Tableau 32.

TABLEAU 32/T.107

Paramètres

Paramètres	Description
OpenApplResult	True: L'application a été identifiée et les objets ont été introduits False: L'application n'a pas été identifiée et aucun objet n'a été introduit. Les paramètres qui ont été utilisés avec la commande VEMMI_Open_Application sont utilisés pour décrire un nouveau cadre d'application

5.3.5 VEMMI_Store_Objects_Resp (réponse à une commande de mémorisation d'objets VEMMI)

Cet élément de service doit être utilisé par le terminal pour répondre à une commande VEMMI_Store_Objects. Voir le Tableau 33.

TABLEAU 33/T.107

Paramètres

Paramètres	Description
StoreResult	True: Les objets ont été mémorisés False: Les objets n'ont pas été mémorisés

5.3.6 VEMMI_Error (erreur VEMMI)

Cet élément de service ne doit être utilisé que par le terminal VEMMI pour signaler à l'application VEMMI différentes situations d'erreur.

Une commande VEMMI_Error avec le paramètre type d'erreur à la valeur «mémoire insuffisante» ne se rapporte qu'à l'objet spécifié. D'autres objets, créés après celui-là, peuvent être mémorisés dans le terminal. Voir le Tableau 34.

TABLEAU 34/T.107

Paramètres

Paramètres	Description
Numéro d'identification d'objet	
ErrorType	
Numéro d'identification de composant	
ErrorComCode	Identification de la commande qui a causé l'erreur

Les types d'erreur suivants sont définis:

- Erreur générale:
une erreur générale s'est produite.
- Commande VEMMI inconnue:
le code de commande VEMMI n'existe pas.
- Commande VEMMI erronée:
la commande VEMMI reçue ne contient pas les paramètres obligatoires ou les contient mais avec des valeurs erronées.
- Erreur de syntaxe d'objet:
la description d'un élément VEMMI n'est pas correcte dans une création d'objet ou une modification de composant.
- Commande VEMMI inattendue:
la commande VEMMI reçue est correcte mais inopportune.
- Mémoire insuffisante:
le terminal n'a pas assez de mémoire pour conserver les données correspondant à une création d'objet ou à une modification de composant.
- Service non pris en charge:
le service demandé n'est pas pris en charge.
- Objet non accepté.
- Type de contenu de données non accepté.
- Indice de couleur non valable.
- Fichier non trouvé.
- Echec de la conversion à la phototrame.
- Indication d'effacement d'objet (un objet fermé a été effacé par le terminal).
- Dépassement de capacité de la mémoire permanente locale.

5.3.7 VEMMI_Resource_Transfer_Abort (abandon de transfert de ressources VEMMI)

Cet élément de service doit être utilisé par le terminal VEMMI pour abandonner une ressource. Voir le Tableau 35.

TABLEAU 35/T.107

Paramètres

Paramètres	Description
TransferID	Identificateur de transfert

6 Introduction d'objets VEMMI

Les objets VEMMI offrent un premier choix d'éléments de dialogue utiles pour les applications VEMMI.

Les objets VEMMI suivants sont définis dans la présente Recommandation:

- barre d'application;
- barre de boutons;
- menu incrusté;
- boîte de dialogue;

- boîte de messages;
- objet opératif;
- objet ressource en phototrame;
- objet ressource en mode vidéotex;
- objet ressource en mode texte;
- objet ressource de police;
- objet de métacode.

On peut désigner une application VEMMI au moyen d'un quelconque des objets ci-dessus. Aucun objet ou composant VEMMI particulier n'est obligatoire dans une application VEMMI. Tous les objets VEMMI peuvent être simultanément présents dans une application VEMMI, sauf la barre d'application. Tous les composants peuvent être simultanément présents dans un objet, sauf que certaines restrictions s'appliquent à des composants particuliers (voir l'article 7).

6.1 La barre d'application

L'objet barre d'application permet à l'utilisateur d'effectuer un choix entre les différentes parties d'application et de sous-application VEMMI offertes par l'application VEMMI sélectionnée. Lorsqu'elle est utilisée, la barre d'application est unique et située soit en haut (barre horizontale) ou à gauche (barre verticale) de la DDA.

6.1.1 Composition

La barre d'application se subdivise en trois groupes logiques de composants de type option de menu. Ces groupes diffèrent par leur comportement et leurs fonctions. Ces trois groupes sont les suivants:

- barre;
- menu déroulant;
- menu empilé.

La barre est une liste, présentée horizontalement ou verticalement, de composants de type option de menu qui représente les différentes parties de l'application VEMMI.

Les menus déroulants sont des listes, présentées verticalement, de sous-composants de type option de menu dépendant d'un même composant option de menu de la barre. Les menus déroulants présentent les différents sous-ensembles de l'application VEMMI.

Les menus empilés sont des listes, présentées verticalement, de composants de type option de menu dépendant d'un même composant option de menu du menu déroulant. Les menus empilés présentent les différents sous-ensembles de l'application VEMMI.

6.2 La barre de boutons

L'objet barre de boutons permet d'effectuer un choix parmi un ensemble d'options, à un moment donné de l'exécution de l'application VEMMI. Chaque option est représentée pour un bouton. La barre de boutons peut être située à n'importe quel endroit de la DDA. Elle peut être horizontale ou verticale.

6.2.1 Composition

La barre de boutons est formée d'une série de composants appelés boutons.

6.3 Le menu incrusté

L'objet menu incrusté offre des options et sous-options appropriées à un élément VEMMI, dans le contexte actuel de cet élément. Cet objet peut s'incruster à n'importe quel endroit de la DDA.

6.3.1 Composition

Le menu incrusté est formé de deux groupes logiques de composants de type option de menu. Ces groupes diffèrent par leur comportement et leurs fonctions. Ces deux groupes sont les suivants:

- menu incrusté primaire;
- menu empilé.

Le menu incrusté primaire est une liste verticale de composants de type option de menu qui offre des options appropriées à un élément VEMMI donné, dans son contexte actuel.

Le menu empilé est une liste verticale de composants de type option de menu dépendant d'un même composant de type option de menu du menu incrusté primaire. Le menu empilé offre des sous-options appropriées à un élément VEMMI, dans son contexte actuel.

6.4 La boîte de dialogue

L'objet boîte de dialogue est celui dans lequel a lieu la principale interaction entre l'utilisateur et l'application VEMMI. Pour que cette interaction soit possible, un ensemble de composants est défini. On peut ranger ces composants en deux catégories: composants de présentation et composants de dialogue.

Les composants de présentation sont inaccessibles; leur objet est uniquement de présenter de manière cohérente et agréable les différents composants de dialogue.

Les composants de dialogue permettent l'interaction entre l'utilisateur et l'application VEMMI.

Cinq composants de présentation sont définis:

- le composant séparateur;
- le composant cadre;
- le composant zone de présentation de texte;
- le composant texte;
- le composant zone d'affichage de graphiques.

Neuf composants de dialogue sont définis:

- le composant poussoir;
- le composant zone de saisie de texte;
- le composant case d'option;
- le composant cercle d'option;
- le composant boîte d'énumération;
- le composant boîte de combinaison;
- le composant ascenseur;
- le composant zone sensible;
- le composant texte sensible.

6.4.1 Composition

6.4.1.1 Le composant séparateur

Le composant séparateur est un trait horizontal ou vertical dont la fonction est d'opérer une démarcation visible entre les différents composants de dialogue situés à l'intérieur de la boîte de dialogue.

6.4.1.2 Le composant cadre

Le composant cadre est un élément de présentation permettant d'opérer une démarcation visible entre une zone particulière de la boîte de dialogue et ses différents composants.

6.4.1.3 Le composant zone de présentation de texte

Le composant zone de présentation de texte est une zone rectangulaire dans laquelle des données en mode texte sont affichées. Les fonctions de ce composant sont les suivantes:

- présenter des informations en mode texte;
- intituler ou étiqueter des composants de dialogue;
- présenter des résultats d'exécution de l'application VEMMI.

6.4.1.4 Le composant texte

La fonction de ce composant est de fractionner des données textuelles abondantes en unités (composants texte) et de définir l'information structurelle nécessaire (concaténation des composants texte) pour l'affichage dans une zone de texte.

6.4.1.5 Le composant zone d'affichage de graphiques

La fonction de ce composant est de présenter des données graphiques à l'utilisateur. Plusieurs formats de codage des données graphiques sont acceptés.

6.4.1.6 Le composant texte sensible

La fonction de ce composant est de définir les opérations d'activation et de validation de chaînes de texte sensible.

6.4.1.7 Le composant poussoir

La fonction du composant poussoir est de déclencher une action locale qui est exécutée immédiatement.

6.4.1.8 Le composant zone de saisie de texte

La fonction du composant zone de saisie de texte est de rassembler des données en mode texte qui ont été saisies par l'utilisateur. Ce composant est une zone rectangulaire formée d'une étiquette en mode texte associée à une zone de saisie.

6.4.1.9 Le composant case d'option

La fonction du composant case d'option est de saisir et d'afficher une option utilisateur indépendante. Ce composant conserve la valeur cochée ou non cochée, quel que soit le contenu d'autres cases d'option.

6.4.1.10 Le composant cercle d'option

La fonction du composant cercle d'option est de saisir et d'afficher une option de l'utilisateur. Ce composant permet d'effectuer un choix unique parmi plusieurs possibilités offertes par une barrette de cercles d'option. Le cochage d'un seul cercle d'option annule le cochage des autres cercles d'option appartenant à la même barrette de cercles d'option.

6.4.1.11 Le composant boîte d'énumération

La fonction du composant boîte d'énumération est d'offrir une ou plusieurs options dans une liste d'items en mode texte. Cette liste n'est généralement pas entièrement visible par l'utilisateur, de sorte que différentes commandes sont proposées pour faire défiler la liste vers le haut ou vers le bas.

6.4.1.12 Le composant boîte de combinaison

La fonction du composant boîte de combinaison est d'associer le rôle d'une boîte d'énumération à choix unique et celui d'une zone de saisie de texte. Ce composant contient une liste d'items en mode texte que l'utilisateur peut dérouler vers le haut ou vers le bas afin de remplir la zone de saisie de texte. Un paramètre du composant boîte de combinaison permet de spécifier si le contenu de la zone de saisie de texte peut ou non être édité. Si c'est le cas, l'utilisateur peut taper son texte directement dans la zone de saisie de texte.

La boîte de combinaison peut se présenter sous forme déroulante, avec des poussoirs. Seuls la zone de saisie de texte et les poussoirs sont affichés au départ. Dès que l'utilisateur coche un poussoir associé à un texte, la boîte d'énumération correspondante est affichée.

6.4.1.13 Le composant ascenseur

Le composant ascenseur permet à l'utilisateur de sélectionner une valeur analogique en déplaçant un pavé de circulation le long d'une barre (horizontale ou verticale) entre une valeur minimale et une valeur maximale. Les intervalles sont déterminés par l'application.

6.4.1.14 Le composant zone sensible

La fonction du composant zone sensible dans la boîte de dialogue est d'offrir une zone de sélection associée à une zone d'affichage.

6.5 Objet opératif

Cet objet permet à une application de désigner un programme qui sera lié à une application VEMMI. Ce type d'objet offre une méthode pour étendre les possibilités d'une application pendant la durée d'exécution.

6.6 Objet ressource en phototrame

Un objet ressource en phototrame contient la définition de la phototrame proprement dite ou simplement un rappel du fichier où figure la définition de la phototrame.

6.7 Objet ressource en mode vidéotex

Un objet ressource en mode vidéotex renferme le contenu en mode vidéotex proprement dit ou simplement un rappel du fichier où figure le contenu en mode vidéotex.

6.8 Objet ressource en mode texte

Cet objet définit le contenu d'un texte comme une ressource qui a pour référence le «numéro d'identification de texte» (TIN). Il renferme le contenu du texte proprement dit ou la référence du fichier où figure ce contenu de texte.

6.9 Objet ressource de police

Cet objet combine un ensemble d'attributs de texte dans une ressource de police qui a pour référence le «numéro d'identification de police» (FIN).

6.10 Objet de métacode

L'objet de métacode contient les commandes VEMMI. Cet objet est commode pour éviter toute phase inutile de dialogue avec l'application assurée par le serveur.

6.11 La boîte de messages

La fonction de l'objet boîte de messages est d'afficher des informations non demandées par un utilisateur mais envoyées par l'application VEMMI en réponse à un événement inattendu ou à une probabilité d'événement indésirable.

7 Description fonctionnelle

7.1 Règles générales pour le comportement des éléments

7.1.1 Interaction avec l'utilisateur

L'utilisateur doit avoir la possibilité d'accéder aux différents éléments VEMMI et de les activer. Le terminal doit également permettre à l'utilisateur de modifier les valeurs des éléments VEMMI et de valider ces saisies.

7.1.2 Actions locales et comptes rendus

Le compte rendu au serveur des saisies d'utilisateur par le terminal doit être provoqué par une commande de compte rendu définie dans une action locale par l'application VEMMI. Durant l'exécution d'une commande de compte rendu, le terminal doit envoyer une commande VEMMI_User_Data à l'application VEMMI. Les événements déclencheurs qui provoquent l'exécution d'actions locales sont les suivants:

- activation d'un composant;
- validation d'un composant.

Les commandes de compte rendu disponibles sont les suivantes:

- signalisation de OIN et CIN;
- signalisation des valeurs actuelles du composant;
- signalisation des valeurs de tous les composants de cet objet;
- signalisation des valeurs des composants de l'objet de rattachement qui ont été modifiées (également par l'intermédiaire d'une commande VEMMI_Modify_Component) depuis le dernier compte rendu ou depuis la création de l'objet s'il s'agit du premier compte rendu.

Les Tableaux 36 et 37 montrent les événements déclencheurs possibles, et les éventuels comptes rendus concernant chaque composant.

La structure globale des actions locales est définie au 4.8.

TABLEAU 36/T.107

Evénements déclencheurs

Composant	Activation	Validation
Barre d'option menu	✓	✓
Option menu déroulant	✓	✓
Option menu empilé	✓	✓
Option menu incrusté	✓	✓
Séparateur d'option menu		
Bouton	✓	✓
Séparateur		
Cadre		
Zone de présentation de texte		
Texte		
Poussoir	✓	✓
Zone de saisie de texte	✓	✓
Case d'option	✓	✓
Cercle d'option	✓	✓
Boîte d'énumération	✓	✓
Boîte de combinaison	✓	✓
Ascenseur	✓	✓
Zone sensible	✓	✓
Texte sensible		
Zone d'affichage de graphiques		

7.1.3 Relations entre objets et composants

L'état fermé d'un objet a priorité sur l'état ouvert d'un composant. Si un objet est fermé, celui-ci, avec tous ses composants, est effacé de l'écran, que ces derniers soient ou non à l'état ouvert. L'état ouvert d'un objet n'a pas priorité sur l'état fermé d'un composant. En cas d'ouverture d'un objet, les composants qui sont à l'état fermé ne sont pas affichés.

Le paramètre d'état inaccessible d'un objet a priorité sur le paramètre d'état accessible d'un composant. Si un objet est mis à l'état inaccessible, tout l'objet, y compris tous ses composants, devient inaccessible, que ces derniers soient ou non accessibles. Le paramètre d'état accessible d'un objet n'a pas priorité sur le paramètre d'état inaccessible d'un composant. Si un objet est mis à l'état accessible, l'utilisateur ne peut entrer en interaction qu'avec les composants accessibles.

TABLEAU 37/T.107

Comptes rendus

Composant	CIN	Valeur de tous les composants	Valeur (Note)
Barre d'option menu	✓		
Option menu déroulant	✓		
Option menu empilé	✓		
Option menu incrusté	✓		
Séparateur d'option menu			
Bouton	✓		
Option menu	✓		
Séparateur			
Cadre			
Zone de présentation de texte			
Texte			
Poussoir	✓	✓	
Zone de saisie de texte	✓	✓	Chaîne
Case d'option	✓	✓	Booléen
Cercle d'option	✓	✓	Booléen
Boîte d'énumération	✓	✓	Chaîne énumérative
Boîte de combinaison	✓	✓	Chaîne
Ascenseur	✓	✓	Entier, booléen
Zone sensible	✓	✓	
Texte sensible			
Zone d'affichage de graphiques			
NOTE – Les entrées de cette colonne spécifient le type de données correspondant aux valeurs dont il est rendu compte.			

7.1.4 Ouverture/fermeture d'objets audio, vidéo, ressource et de métacode

En cas d'ouverture d'un objet audio, celui-ci doit produire un son. Une fois ce son produit, l'objet est considéré comme étant fermé.

La même règle s'applique aux objets vidéo. En cas d'ouverture de l'objet, la visualisation commence et lorsqu'elle prend fin l'objet se ferme.

L'opération d'ouverture d'une ressource vise simplement à vérifier si celle-ci est présente dans le terminal.

L'opération d'ouverture permet d'assurer le traitement d'un objet de métacode. A la fin du traitement, cet objet est considéré comme étant ouvert.

7.1.5 Opération d'agrandissement plein écran

NOTE – L'opération décrite dans le présent paragraphe n'est pas obligatoire pour un terminal VEMMI. Un terminal qui ne peut exécuter cette opération ne tiendra pas compte des attributs et opérations correspondants.

Un objet peut être défini comme pouvant être agrandi à la taille maximale de l'écran. On utilisera à cet effet le bouton d'affichage plein écran figurant dans la barre de titre de cet objet lequel, après validation de ce bouton, sera affiché sur toute la surface de l'écran. Un bouton de remise à l'état initial complète l'objet agrandi lequel, après validation de ce bouton, sera réaffiché dans sa taille initiale. Ces opérations dépendent du terminal et de l'interface utilisateur utilisés, en particulier la taille agrandie. L'application assurée par le serveur peut uniquement signaler qu'un objet peut être agrandi à la taille maximale de l'écran.

Les modalités d'agrandissement de tous les éléments et composants d'objets, à l'exception du composant zone de présentation de texte, sont fonction du terminal utilisé. Le composant zone de présentation de texte d'une boîte de dialogue se comporte de manière à permettre à un objet agrandi d'étoffer localement le contenu en mode texte dans sa zone de visualisation. Si le composant zone de présentation de texte contient dans sa taille initiale la totalité du texte, l'opération d'agrandissement plein écran dépend du terminal utilisé. Si le texte n'est affiché qu'en partie dans sa position initiale, le terminal doit ajouter un texte complémentaire dans l'objet agrandi à toute la surface de l'écran. L'objet agrandi peut afficher le texte sous une forme reformatée. Celle-ci peut être définie par le serveur.

Si l'objet est fermé dans sa taille agrandie et par la suite réouvert, il est affiché dans sa taille initiale (non agrandie). Si un objet agrandi à toute la surface de l'écran est désactivé, il conserve sa taille.

D'autres opérations locales relatives aux objets, non définissables par le serveur, peuvent être offertes à l'utilisateur en fonction du terminal utilisé, à savoir: réduction à la taille minimale, déplacement, réduction en icône, élargissement ou rétrécissement linéaires, etc.

7.1.6 Conventions applicables pour les notations

Les conventions applicables aux paragraphes ci-dessous pour les notations sont les suivantes:

- 1) dans les paragraphes ci-dessous du présent article, la description de chaque objet et de chaque composant contient une liste d'attributs. Ces listes indiquent les attributs qui sont nécessaires pour décrire les propriétés des éléments. La syntaxe précise de codage des éléments est définie dans l'article 9;
- 2) les numéros d'identification des objets ressources, BIN (numéro d'identification de phototrame), FIN (numéro d'identification de police), TIN (numéro d'identification de texte), VIN (numéro d'identification de mode vidéotex) sont des cas particuliers de OIN, utilisés sur le même niveau, par exemple comme paramètres des commandes VEMMI. Différent de ces numéros, le CIN comporte toujours l'indication d'un composant.

Les conventions de dénomination applicables aux attributs booléens de la description des éléments VEMMI obéissent aux règles suivantes:

- si l'attribut est présent dans le codage, sa valeur booléenne correspond à la valeur exprimée par son nom.

Exemple

La valeur par défaut de l'état «ouvert est true». Si l'attribut «fermé» est présent dans le codage d'un élément VEMMI, cet élément doit être à l'état «fermé». Si l'attribut «fermé» n'est pas présent dans le codage d'un élément VEMMI, cet élément doit être à son état par défaut, l'état «ouvert».

7.1.7 Mnémonique

Lorsque des données en mode texte sont utilisées dans un composant, le texte peut contenir le caractère «&» qui indique que le caractère suivant est une clé mnémonique. Celle-ci peut être utilisée pour valider ou activer le composant. Pour présenter le caractère «&» dans un texte, on doit le doubler (exemple: contenu de texte = «&Cats && Dogs»; texte présenté à l'utilisateur «Cats & Dogs»).

7.2 Formats de texte

7.2.1 Texte de haute qualité VEMMI

L'utilisation d'«attributs dans le texte» permet d'appliquer des attributs de texte [changement de police de caractères, du corps des caractères, etc., à l'intérieur d'une chaîne de caractères (chaîne de texte)] et de définir des zones sensibles de texte [texte sensible ou «petit groupe d'instructions» (hot-spot)]. Ces attributs sont applicables aux éléments suivants:

- Composants zone de présentation de texte d'une boîte de dialogue.
- Etiquettes.
 - 1) *Attributs de texte*

La police attribuée peut être modifiée au cours d'une opération d'affichage en mode texte. La police est désignée par FIN. En cas de choix d'une nouvelle police, celle-ci s'applique à tout le reste du texte jusqu'au nouvel attribut in-text modifiant la police ou jusqu'à la fin du composant. La police attribuée au départ au texte du composant est définie comme étant un attribut supplémentaire du composant.

L'exemple ci-dessous illustre l'utilisation d'attributs in-text. Les guillemets <...> désignent les attributs des composants avec leur paramètre.

Exemple:

....

<texte1> = «Note de service pour l'utilisateur: (CR,LF,CR,LF)»

<utiliser FIN 5>

<texte2> = «Les bandes de négatifs de taille 110»

<utiliser FIN 6>

<texte3> = «ne doivent pas être coupées mais nous être retournées telles quelles pour retraitage.»

<utiliser FIN 5>

<texte4> = «Nous ne pouvons pas tirer d'épreuves à partir de bandes de négatifs coupées.»

<utiliser FIN 7>

<texte5> = «(CR,LF,CR,LF) Emis par l'A.P.L.»

.....

FIN 5 est défini comme suit: ROMAN, hauteur 12

FIN 6 est défini comme suit: ROMAN, hauteur 12, gras, souligné

FIN 7 est défini comme suit: SWISS, hauteur 8

La séquence de texte ci-dessus, avec ses attributs in-text, pourrait se présenter comme suit:

Note de service pour l'utilisateur:

Les bandes de négatifs de taille 110 *ne doivent pas être coupées mais nous être retournées telles quelles pour retraitage*. Nous ne pouvons pas tirer d'épreuves à partir de bandes de négatifs coupées.

Emis par l'A.P.L.

2) *Attributs in-text pour la définition du texte sensible*

Chaque chaîne de caractères (chaîne de texte) définie comme étant sensible se rapporte à un composant texte sensible. Ce composant contient deux attributs qui définissent les opérations d'activation et de validation à exécuter au moment où l'utilisateur entre en interaction avec le texte sensible.

On peut mélanger les deux groupes d'attributs dans un même composant de texte, de manière que l'application puisse appliquer des attributs de texte (couleur, par exemple) aux chaînes sensibles. Une telle chaîne peut apparaître en vert et éventuellement aussi en surbrillance, de manière à se différencier facilement du texte en clair, qui peut se présenter en caractères noirs sur fond blanc. Ces attributs permettent à l'application d'offrir à l'utilisateur un affichage de type «hypertexte» avec VEMMI.

7.2.2 **Étiquettes et titres de texte**

Les contenus en mode texte des étiquettes et des titres sont coupés sur les bordures de leurs contenants. Cela s'applique aux attributs suivants:

- tous les attributs de titre;
- tous les attributs d'étiquette;
- attribut de contenu en mode texte des composants option de menu;
- attribut de texte d'énumération des composants boîte d'énumération et boîte de combinaison.

7.3 La barre d'application

La barre d'application se subdivise en trois groupes logiques de composants de type option de menu. Ces groupes diffèrent par leur comportement et leurs fonctions. Ces trois groupes sont les suivants:

- barre;
- menu déroulant;
- menu empilé.

NOTE 1 – La représentation verticale de la barre d'application sur un terminal VEMMI constitue une caractéristique facultative. Si l'attribut vertical est présent avec la valeur true dans le codage d'une barre d'application, le terminal peut ne pas tenir compte de cet attribut et présenter la barre d'application horizontalement. Toutefois, aucun message d'erreur ne doit être envoyé au serveur.

NOTE 2 – Les attributs de positionnement et de dimensionnement d'une barre d'application et de ses différents composants de type option de menu sont facultatifs. Si des attributs de positionnement ou de dimensionnement sont définis dans le codage d'une barre d'application, le terminal peut ne pas tenir compte de ces attributs et présenter la barre d'application en la faisant commencer dans le coin supérieur gauche de la DDA et en lui donnant la dimension voulue compte tenu de la longueur du texte à insérer. Toutefois, aucun message d'erreur ne doit être envoyé au serveur.

La barre est une liste, présentée horizontalement ou verticalement, de composants de type option de menu. Les menus déroulants sont des listes, présentées verticalement, de sous-composants de type option de menu dépendant d'un même composant option de menu de la barre. Les menus empilés sont des listes, présentées verticalement, de composants de type option de menu dépendant d'un même composant option de menu du menu déroulant.

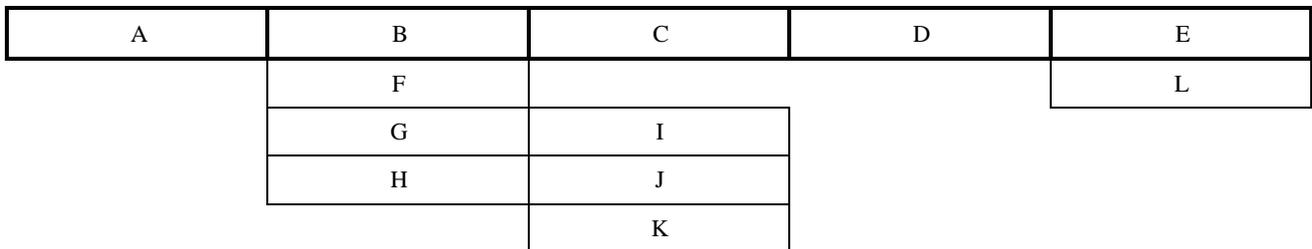
Les composants de type option de menu de la barre d'application ne doivent contenir que du texte.

Un composant séparateur d'option de menu peut être utilisé pour séparer visuellement les différents composants d'option de menu d'un groupe logique. On définit ce composant en sélectionnant l'attribut «Separator» avec la définition d'une option de menu ordinaire.

La structure de l'objet barre d'application est indiquée par l'ordre canonique décroissant de ses composants (de gauche à droite avec priorité à la profondeur).

Dans l'exemple de barre d'application ci-dessous, l'ordre canonique décroissant des composants est le suivant:

(AB(FG(IJK)H)CDE(L))

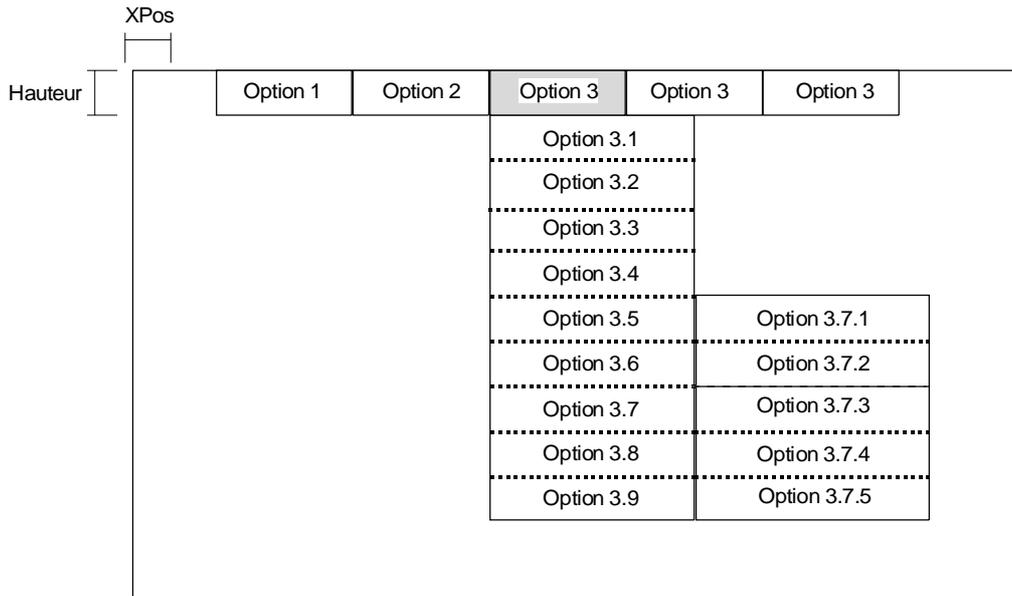


NOTE 3 – Dans l'exemple ci-dessus, les parenthèses servent uniquement à indiquer les différents niveaux d'encapsulation des différents groupes d'objets, qui déterminent l'ordre à suivre pour la description.

La description de la structure de l'objet est donc la suivante:

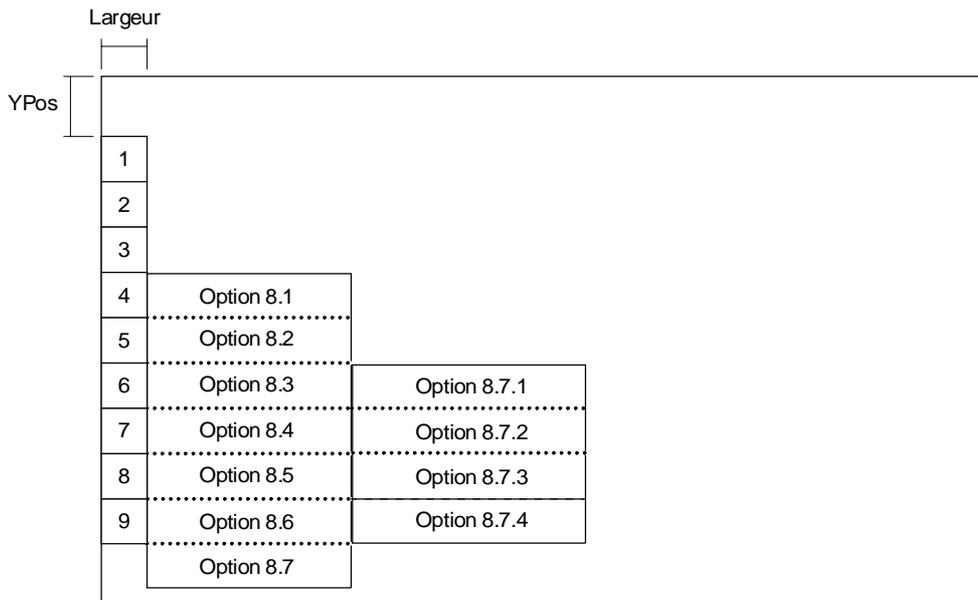
- objet: barre d'application;
- composant: barre d'option de menu A;
- composant: barre d'option de menu B;
- composant: option de menu déroulant F;
- composant: option de menu déroulant G;
- composant: option de menu empilé I;
- composant: option de menu empilé J;
- composant: option de menu empilé K;
- composant: option de menu déroulant H;
- composant: barre d'option de menu C;
- composant: barre d'option de menu D;
- composant: barre d'option de menu E;
- composant: option de menu déroulant L.

Aspect visuel général (voir les Figures 7 et 8)



T0823880-95/d07

FIGURE 7/T.107



T0823890-95/d08

FIGURE 8/T.107

Attributs

- Vertical: la barre d'application doit être présentée verticalement (voir la Note 1).
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément. Il n'est applicable qu'aux barres d'application horizontales (voir la Note 2).
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément. Il n'est applicable qu'aux barres d'application verticales (voir la Note 2).
- Height: cet attribut permet d'indiquer la hauteur de la barre. Il n'est applicable qu'aux barres d'application horizontales (voir la Note 2).
- Width: cet attribut permet d'indiquer la largeur de la barre. Il n'est applicable qu'aux barres d'application verticales (voir la Note 2).

- FirstActive: cet attribut permet d'indiquer le CIN de l'option de menu qui est active par défaut, la première fois que l'objet est ouvert.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'objet n'est pas accessible.

7.3.1 Composition

7.3.1.1 Composants de type option de menu dans la barre

Description

La barre est une liste de composants de type option de menu successifs, positionnés les uns à côté des autres, horizontalement ou verticalement.

Représentation horizontale:

- la barre d'application doit être affichée en haut de la DDA;
- le nombre de rangées attribuées à chaque option de menu doit être le même.

Représentation verticale:

- la barre d'application doit être affichée à gauche de la DDA;
- le nombre de colonnes attribuées à chaque option de menu doit être le même.

Comportement

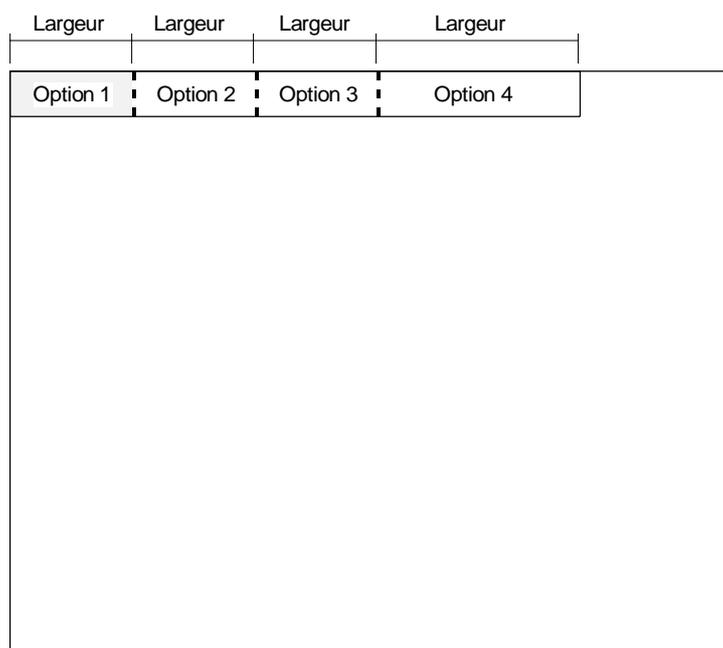
Lorsque la barre d'application est active, le terminal doit accentuer une option de menu et offrir à l'utilisateur des possibilités de recherche par décalage et de validation. Une touche mnémonique permet de valider l'option de menu (voir 7.1.7).

Si une option de menu de la barre est validée, l'action locale associée doit être exécutée et le menu déroulant associé doit être ouvert (l'option reste accentuée).

Fonctions d'interactivité

- Décalage (de recherche).
- Validation.

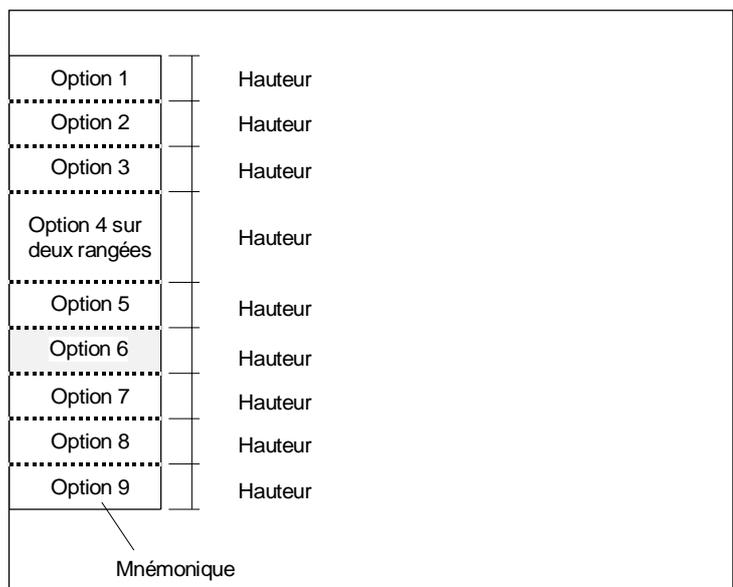
Aspect visuel (voir les Figures 9 et 10)



T0815510-94/d09

FIGURE 9/T.107

Barre de menu horizontale



T0815520-94/d10

FIGURE 10/T.107

Barre de menu verticale

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- Height: cet attribut permet d'indiquer la hauteur du composant. Il n'est applicable qu'aux barres verticales (voir 7.3, Note 2).
- Width: cet attribut permet d'indiquer la largeur du composant. Il n'est applicable qu'aux barres horizontales (voir 7.3, Note 2).
- NotAccessible: l'élément n'est pas accessible.
- Text: cet attribut permet d'indiquer le contenu en mode texte du composant.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.3.1.2 Composants de type option de menu dans le menu déroulant

Description

Le menu déroulant est une liste de composants de type option de menu consécutifs, associés aux mêmes options de menu de la barre et présentés verticalement sur plusieurs rangées.

Le nombre de colonnes attribuées à chaque option de menu doit être le même.

Le menu déroulant doit être positionné au-dessous ou à côté de l'option de menu dont il dépend sur la barre.

Dans un menu déroulant, un séparateur permettant de délimiter clairement les différentes options de menu peut être inséré entre celles-ci.

Comportement

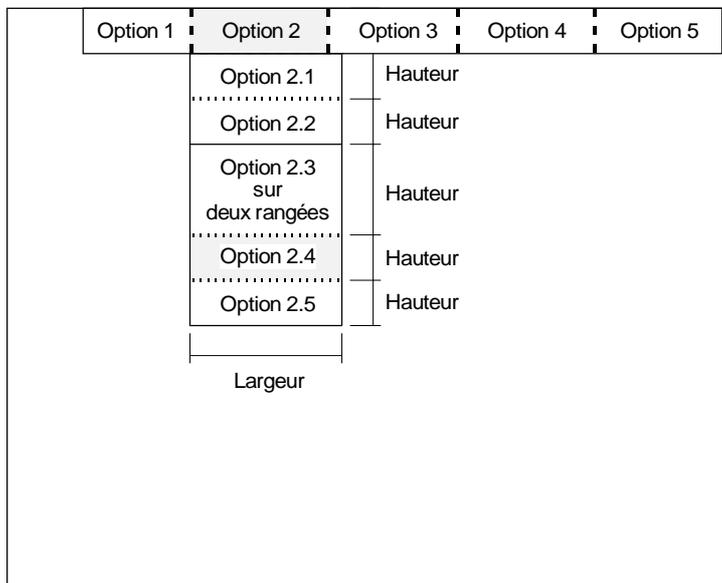
Lorsque le menu déroulant est actif, le terminal doit accentuer une option de menu et offrir à l'utilisateur des possibilités de recherche par décalage et de validation. Une touche mnémonique permet de valider l'option de menu (voir 7.1.7).

En cas de validation d'une option de menu, il faut exécuter l'action locale associée puis ouvrir le menu empilé (l'option de menu reste accentuée). Dans le cas où aucun menu empilé n'est associé à l'option de menu après une validation par l'utilisateur, il faut exécuter l'action locale associée puis fermer le menu déroulant.

Fonctions d'interactivité

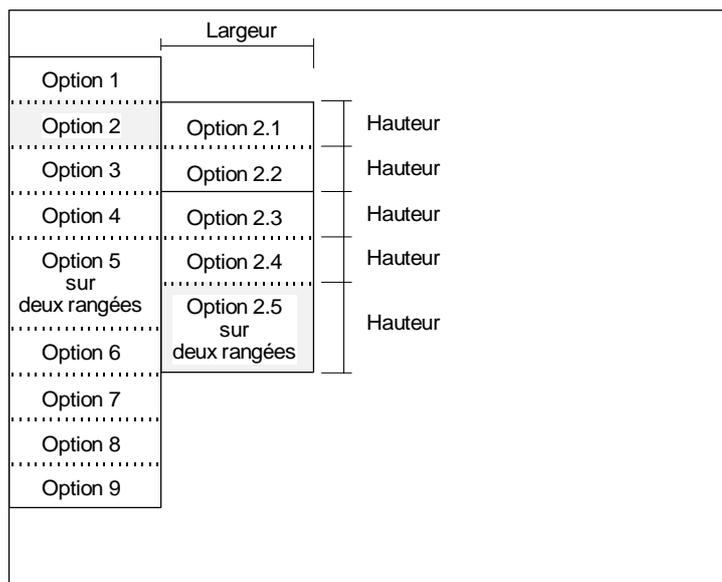
- Décalage (de recherche).
- Validation.

Aspect visuel (voir les Figures 11 et 12)



T0824320-95/d11

FIGURE 11/T.107
Menu déroulant



T0824330-95/d12

FIGURE 12/T.107
Menu déroulant

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- Height: cet attribut permet d'indiquer la hauteur du composant (voir 7.3, Note 2).
- Width: cet attribut permet d'indiquer la largeur du composant. Sa valeur doit être égale à toutes les options du menu déroulant (voir 7.3, Note 2).
- NotAccessible: l'élément n'est pas accessible.
- Text: cet attribut permet d'indiquer le contenu en mode texte du composant.
- Separated: un séparateur doit être dessiné entre les éléments.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.3.1.3 Composants de type option de menu dans le menu empilé

Les composants de type option de menu du menu empilé sont communs aux objets menu déroulant et menu incrusté.

Description

Le menu empilé est une liste de composants de type option de menu consécutifs, associés aux mêmes options du menu déroulant ou incrusté et présentés verticalement sur plusieurs rangées.

Le nombre de colonnes attribuées à chaque option de menu doit être le même.

Le menu empilé doit être positionné à côté du menu déroulant ou incrusté dont il dépend.

Dans un menu empilé, un séparateur permettant de délimiter clairement les différentes options de menu peut être inséré entre celles-ci.

La Figure 13 montre la présentation recommandée pour un menu empilé. Si le terminal ne peut pas présenter un menu empilé comme indiqué à la Figure 13, il peut, à titre de solution de remplacement, présenter les options du menu empilé dans le menu déroulant, mais il doit établir une correspondance visible entre les options du menu empilé et les options correspondantes du menu déroulant. Le terminal doit mettre l'option du menu déroulant, dont dépend le menu empilé, à l'état inaccessible et la présenter sous forme de titre pour les options du menu empilé (voir la Figure 14).

Comportement

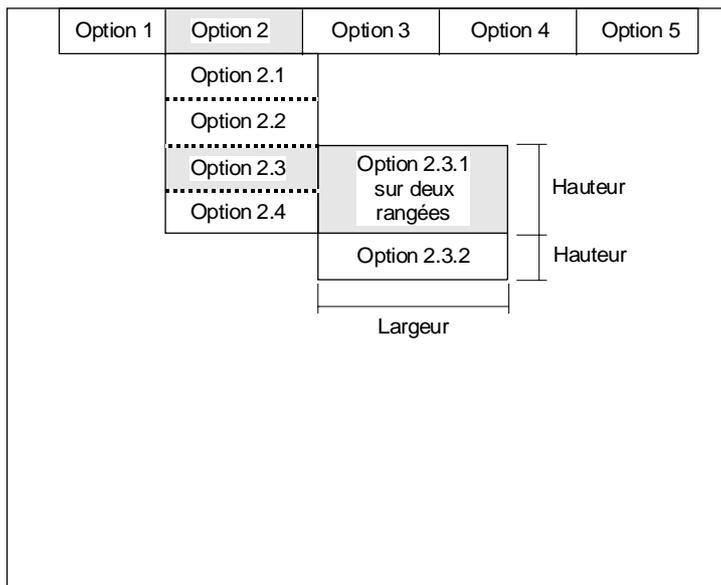
Lorsque le menu empilé est activé, le terminal doit accentuer une seule option de menu et offrir à l'utilisateur des possibilités de recherche par décalage et de validation. Une touche mnémonique permet de valider l'option de menu (voir 7.1.7).

Après la validation d'une option de menu, il convient d'exécuter l'action locale correspondante puis de fermer le menu empilé et le menu déroulant (menu incrusté).

Fonctions d'interactivité

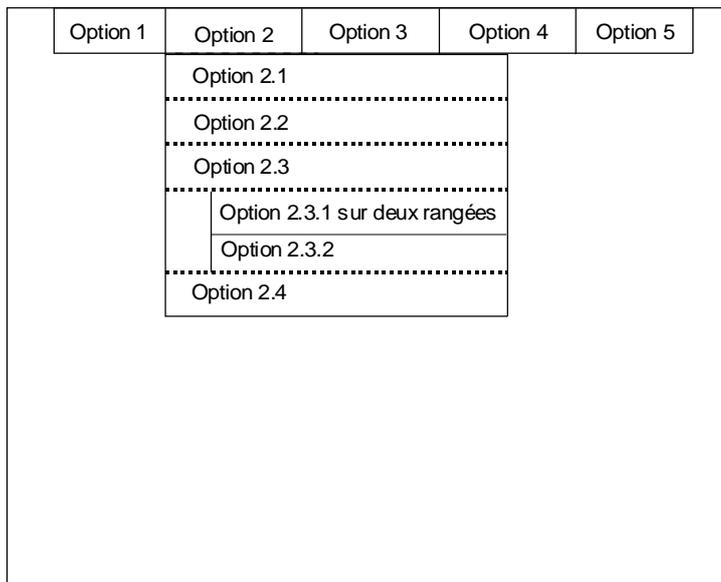
- Décalage (de recherche).
- Validation.

Aspect visuel (voir les Figures 13 et 14)



T0823920-95/d13

FIGURE 13/T.107
Menu empilé (présentation recommandée)



T0823930-95/d14

FIGURE 14/T.107
Menu empilé (présentation recommandée)

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- Height: cet attribut permet d'indiquer la hauteur du composant (voir 7.3, Note 2).
- Width: cet attribut permet d'indiquer la largeur du composant. Sa valeur doit être égale à toutes les options du menu empilé (voir 7.3, Note 2).
- NotAccessible: l'élément n'est pas accessible.

- Text: cet attribut permet d'indiquer le contenu en mode texte du composant.
- Separated: un séparateur doit être dessiné entre les éléments.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.4 La barre de boutons

Description

La barre de boutons est une liste de boutons consécutifs, positionnés les uns à côté des autres dans le sens horizontal ou vertical.

Représentation horizontale:

- le nombre de rangées attribuées à chaque bouton doit être le même.

Représentation verticale:

- le nombre de colonnes attribuées à chaque bouton doit être le même.

NOTE – Si les attributs «Height» et «Width» sont présents dans la définition d'une barre de boutons, tous les boutons de celle-ci doivent avoir les mêmes dimensions. Dans ce cas, la présence des attributs de dimensionnement des composants de la barre de boutons n'est pas nécessaire. Si seul l'attribut «Height» est présent dans la définition d'une barre de boutons horizontale, il indique la hauteur de tout l'objet. La largeur des composants est alors indiquée pour chaque composant. Si seul l'attribut «Width» est présent dans la définition d'une barre de boutons verticale, il indique la largeur de tout l'objet. La hauteur des composants est alors indiquée pour chaque composant.

Comportement

Lorsque la barre de boutons est active, le terminal doit accentuer un seul bouton et offrir à l'utilisateur des possibilités de recherche par décalage et de validation.

Fonctions d'interactivité

- Décalage (de recherche).
- Validation.

Aspect visuel (voir la Figure 15)

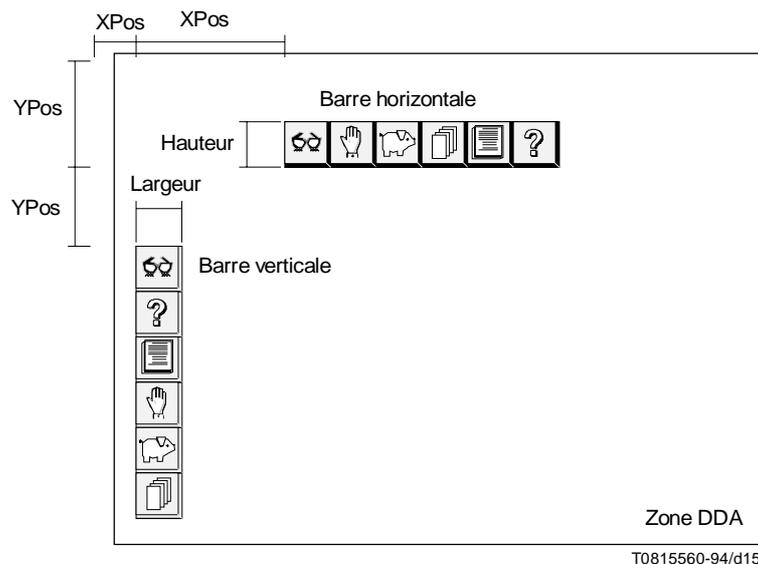


FIGURE 15/T.107

Barre de boutons

Attributs

- XPos: cet attribut permet d'indiquer – en coordonnées normées d'appareil (NDC) – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Vertical: commutateur booléen entre les deux options.
- Height: cet attribut permet d'indiquer la hauteur de la barre de boutons. Il n'est applicable qu'aux barres de boutons horizontales (voir la Note).
- Width: cet attribut permet d'indiquer la largeur de la barre de boutons. Il n'est applicable qu'aux barres de boutons verticales (voir la Note).
- FirstActive: cet attribut permet d'indiquer le CIN du bouton qui est actif par défaut, la première fois que l'objet est ouvert.
- Modal: l'élément doit être modal.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.

7.4.1 Composition

7.4.1.1 Le composant bouton

Description

Le composant bouton se présente sous la forme d'une fenêtre rectangulaire dans la DDA.

S'il est formé de données en mode texte, il doit être représenté par une étiquette de texte, éventuellement associée à un graphe dessinant la forme du bouton. Cette forme graphique doit être insérée dans l'espace attribué par l'application VEMMI pour l'ensemble du composant bouton. Si la forme graphique ne remplit pas tout l'espace attribué pour le bouton, le terminal peut en redimensionner le contenu. Le terminal peut centrer les données en mode texte dans la zone d'affichage élémentaire qui est réservée par l'application VEMMI pour le composant.

Le dessin du bouton et l'éventuel effet d'enfoncement dépendent du terminal.

Comportement

Lorsqu'un bouton est actif, le terminal doit l'accentuer. Une touche mnémonique permet de valider le bouton (voir 7.1.7).

Lorsqu'un effet d'enfoncement est mis en œuvre, le bouton doit revenir à son état d'affichage initial après avoir été validé par l'utilisateur.

Fonctions d'interactivité

- Activation.
- Validation.

Aspect visuel (voir la Figure 16)

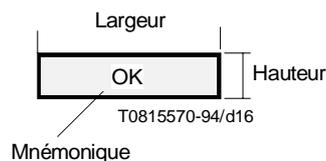


FIGURE 16/T.107

Bouton

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- Height: cet attribut permet d'indiquer la hauteur du composant. Il n'est applicable qu'aux boutons des barres de boutons verticales (voir 7.4, Note).
- Width: cet attribut permet d'indiquer la largeur du composant. Il n'est applicable qu'aux boutons des barres de boutons horizontales (voir 7.4, Note).
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément est inaccessible.
- BIN: numéro d'identification de phototrame (BIN) du composant.
- Text: contenu en mode texte du composant.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.5 Le menu incrusté

Le menu incrusté se subdivise en deux groupes logiques de composants de type option de menu. Ces groupes diffèrent par leur comportement et leurs fonctions. Ces deux groupes sont les suivants:

- menu incrusté primaire;
- menu empilé.

Le menu incrusté primaire est une liste, présentée verticalement, de composants de type option de menu. Le menu empilé est une liste, présentée verticalement, de composants de type option de menu dépendant d'un même composant option de menu du menu incrusté primaire.

Un séparateur d'option de menu peut être utilisé pour séparer visuellement les différents composants d'option de menu d'un groupe logique. On définit ce séparateur en sélectionnant l'attribut «Separator» avec la définition d'une option de menu ordinaire.

La structure de l'objet menu incrusté est indiquée, avec les conventions suivantes, par l'ordre canonique décroissant de ses composants (de haut en bas).

Dans l'exemple de menu incrusté ci-dessous, l'ordre canonique décroissant des composants est le suivant:

(BFG(IJK)H).

B	
F	
G	I
H	J
	K

NOTE 1 – Dans l'exemple ci-dessus, les parenthèses servent uniquement à indiquer les différents niveaux d'encapsulation des différents groupes d'objets, qui déterminent l'ordre à suivre pour la description.

La description de la structure de l'objet est donc la suivante:

- objet: menu incrusté;
- composant: option de menu incrusté B;
- composant: option de menu incrusté F;
- composant: option de menu incrusté G;
- composant: option de menu empilé I;

- composant: option de menu empilé J;
- composant: option de menu empilé K;
- composant: option de menu incrusté H.

Fonctions d'interactivité

- Décalage (de recherche).
- Validation.

Aspect visuel général (voir la Figure 17)

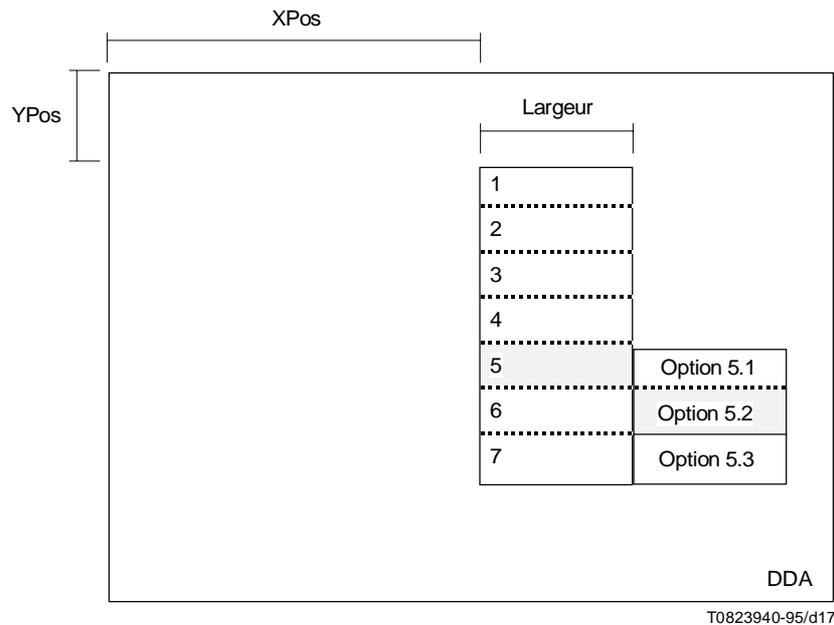


FIGURE 17/T.107

Un titre en mode texte peut être donné au menu incrusté. Ce titre sera affiché sur la première rangée du menu incrusté primaire.

NOTE 2 – Les attributs de dimensionnement des composants option de menu du menu incrusté sont facultatifs. Si des attributs de dimensionnement sont définis dans le codage d'un menu incrusté, le terminal peut ne pas tenir compte de ces attributs et donner au titre la dimension voulue compte tenu de la longueur du texte à insérer. Toutefois, aucun message d'erreur ne doit être envoyé au serveur.

Attributs

- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du menu incrusté (voir la Note 2).
- Title: cet attribut permet d'indiquer le titre de l'objet. Ce titre doit être affiché sur la première rangée de l'objet.
- TitleFont: cet attribut permet d'indiquer le FIN du titre.
- FirstActive: cet attribut permet d'indiquer le CIN du composant qui est actif par défaut, la première fois que l'objet est ouvert.
- Modal: l'élément doit être modal.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.

7.5.1 Composition

7.5.1.1 Composants de type option de menu du menu incrusté primaire

Description

Le menu incrusté primaire est une liste de composants de type option de menu consécutifs, présentés verticalement sur plusieurs lignes.

Le nombre de colonnes attribuées à chaque option de menu doit être le même.

Dans un menu incrusté primaire, un séparateur permettant de délimiter clairement les différentes options de menu peut être inséré entre celles-ci.

Comportement

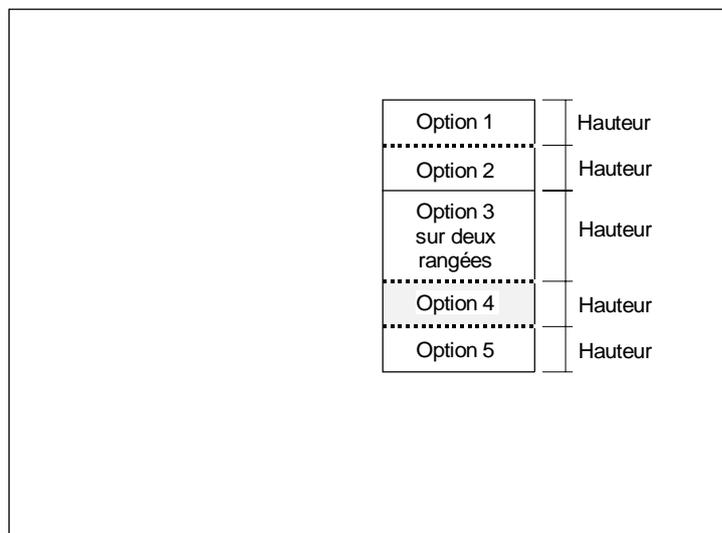
Lorsque le menu incrusté primaire est activé, le terminal doit accentuer une option de menu et offrir à l'utilisateur des possibilités de recherche par décalage et de validation. Une touche mnémonique permet de valider l'option de menu (voir 7.1.7).

En cas de validation d'un composant de type menu incrusté, il faut exécuter l'action locale associée puis ouvrir le menu empilé correspondant (le composant de type menu incrusté reste accentué). Dans le cas où aucun menu empilé n'est associé à l'option de menu après une validation par l'utilisateur, il faut exécuter l'action locale associée puis fermer le menu incrusté.

Fonctions d'interactivité

- Décalage (de recherche).
- Validation.

Aspect visuel (voir la Figure 18)



T0823950-95/d18

FIGURE 18/T.107

Menu incrusté

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- Height: cet attribut permet d'indiquer la hauteur du composant (voir 7.5, Note 2).
- Closed: l'élément doit être à l'état fermé.

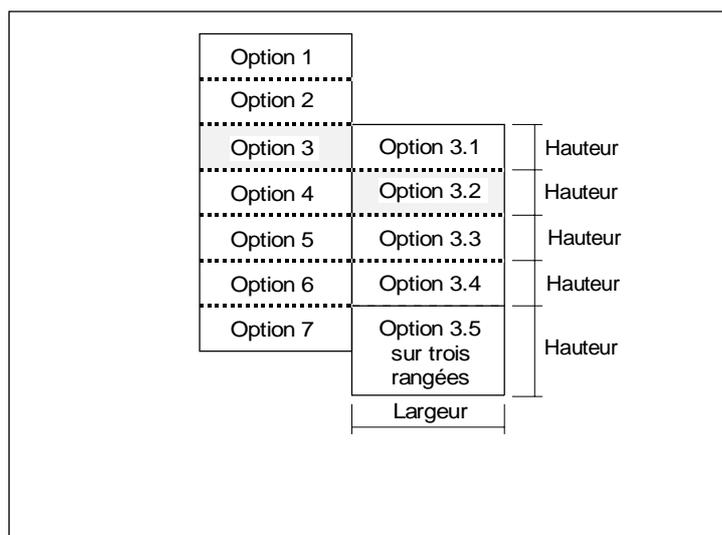
- NotAccessible: l'élément n'est pas accessible.
- Text: cet attribut permet d'indiquer le contenu en mode texte du composant.
- Separated: les éléments doivent être séparés.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.5.1.2 Composants de type option de menu du menu empilé

Description, comportement, fonctions d'interactivité, attributs

Voir 7.3.1.3.

Aspect visuel (voir la Figure 19)



T0823960-95/d19

FIGURE 19/T.107

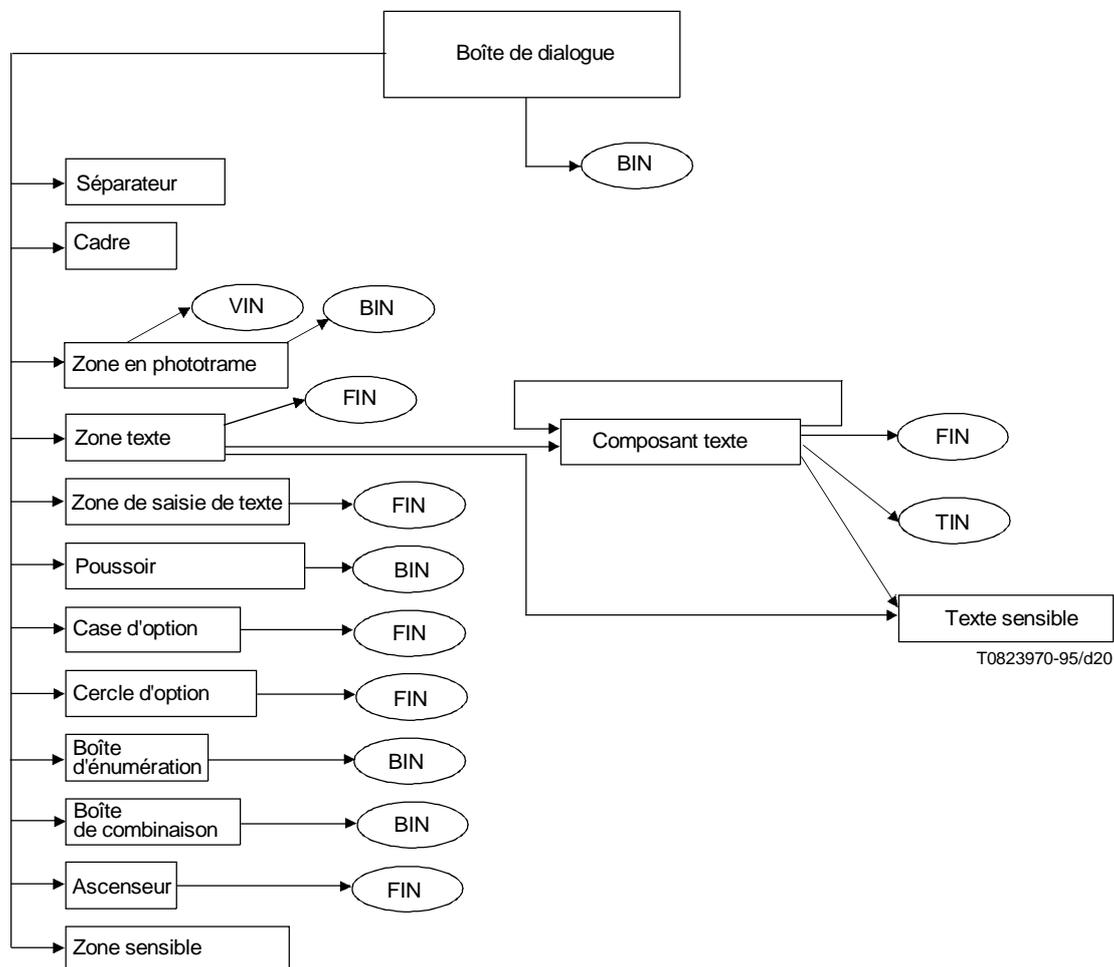
Présentation recommandée d'un menu incrusté avec menu empilé associé

7.6 La boîte de dialogue

Description

L'objet boîte de dialogue est une fenêtre rectangulaire dans la DDA qui contient des composants VEMMI afin d'établir une interactivité avec l'utilisateur.

Les composants proprement dits contiennent d'autres composants ou objets ressources (ou y renvoient). La Figure 20 montre les relations entre les éléments à l'intérieur de la boîte de dialogue. Chaque flèche renvoie à un composant ou à un objet ressource. Ces renvois peuvent s'appliquer à plusieurs composants ou objets ressources, de même que la boîte de dialogue peut contenir plusieurs composants du même type.



T0823970-95/d20

FIGURE 20/T.107
Relations entre les éléments d'une boîte de dialogue

Un titre en mode texte peut être donné à la boîte de dialogue; il doit figurer sur la première rangée de celle-ci. La hauteur du titre est égale à 1/32 NDC.

La boîte de dialogue peut être enrichie d'une bordure dont l'épaisseur sera d'une position de caractère. Si elle est présente, cette bordure doit être incluse dans les dimensions définies par l'application VEMMI. Si une bordure est demandée, un cadre doit être dessiné par le terminal. Le dessin de ce cadre dépend du terminal (mais la largeur du cadre ne doit pas dépasser 1/160 NDC).

Le point d'origine des objets de la boîte de dialogue est le coin supérieur gauche du rectangle constitué par la boîte, que celle-ci comporte ou non un titre. Les composants sont positionnés par rapport à ce coin pour les boîtes de dialogue ne comportant pas de titre. Dans le cas de boîtes avec titre, ce point est décalé vers le bas d'une distance correspondant à la hauteur du titre et à la largeur des bordures et, horizontalement, d'une distance égale à la largeur des bordures.

La zone de fond de la boîte de dialogue est le rectangle matérialisé par les dimensions de la boîte, moins la barre de titre et la bordure. Le fond peut être d'une couleur uniforme ou contenir des données.

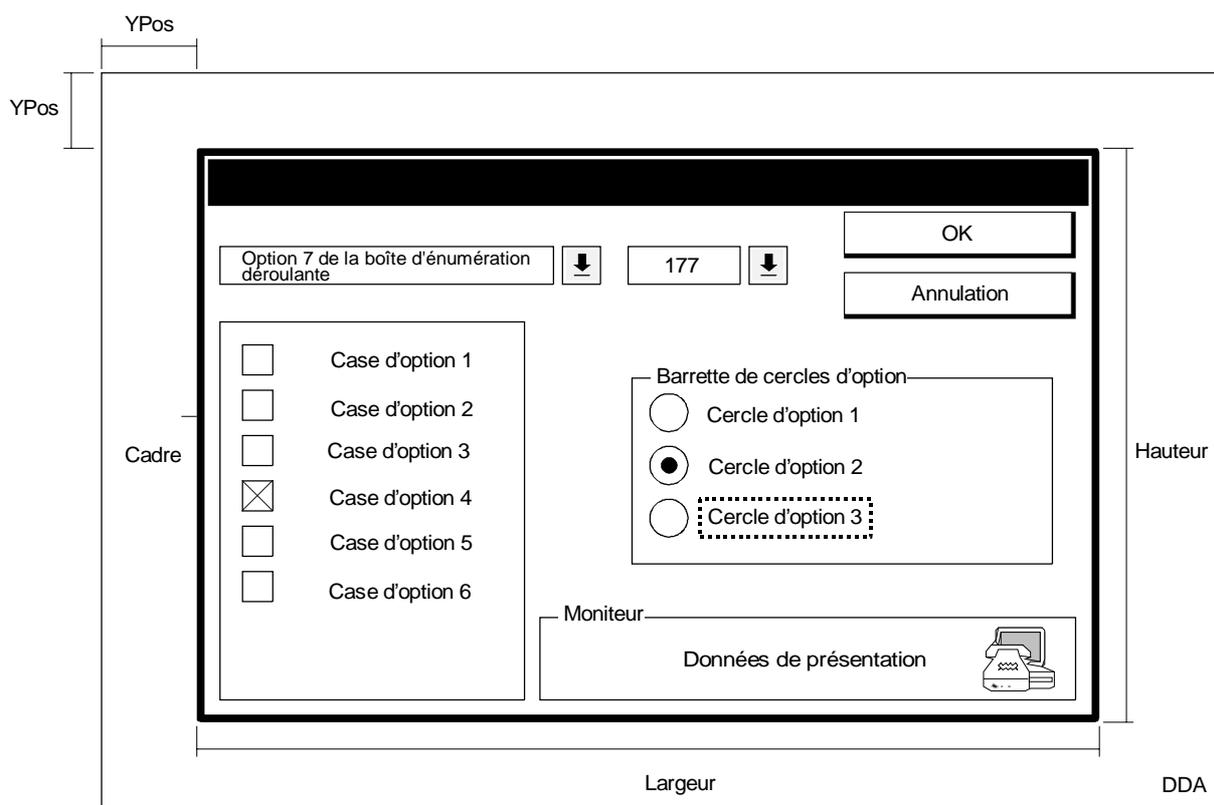
Comportement

L'utilisateur peut activer les composants à l'aide d'un pointeur ou du clavier. S'il utilise à cette fin les touches du clavier qui assurent les fonctions «page précédente» et «page suivante», les composants de la boîte de dialogue doivent être activés dans l'ordre où ils sont reçus.

Fonction d'interactivité

- Déplacement.

Aspect visuel (voir la Figure 21)



T0815610-94/d21

FIGURE 21/T.107

Boîte de dialogue

Attributs

- **XPos:** cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- **YPos:** cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- **Width:** cet attribut permet d'indiquer la largeur de l'objet.
- **Height:** cet attribut permet d'indiquer la hauteur de l'objet.
- **NoBorder:** aucune bordure ne doit être dessinée.
- **Title:** cet attribut permet d'indiquer le titre de l'objet. Ce titre doit être affiché sur la première rangée de l'objet.

- FirstActive: cet attribut permet d'indiquer le CIN du composant qui est actif par défaut, la première fois que l'objet est ouvert.
- Modal: l'élément doit être modal.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.
- StoreInitialValues: le terminal doit mémoriser les valeurs initiales du composant au moment où l'objet est créé. Si une action locale «rétablir valeurs initiales» est appliquée, les valeurs mémorisées doivent être rétablies.
- Maximizable: l'objet peut être agrandi à la taille maximale de l'écran.
- Colour: indice de couleur du fond.
- BIN: numéro d'identification de phototrame pour le remplissage du fond de la boîte de dialogue.
- DispType: cet attribut spécifie le mode d'affichage en phototrame du fond: centré, élargi (par défaut) ou en juxtaposé.

7.6.1 Composition

7.6.1.1 Le composant séparateur

Description

Le composant séparateur est un trait horizontal ou vertical dont la fonction est d'opérer une démarcation visible entre les différents composants de dialogue situés à l'intérieur d'une boîte de dialogue.

Comportement

Le composant séparateur doit être inaccessible.

Fonctions d'interactivité

- Néant.

Aspect visuel (voir la Figure 22)

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Vertical: l'élément doit être dessiné verticalement.
- Height: cet attribut permet d'indiquer la hauteur du composant. Il n'est applicable qu'aux séparateurs verticaux.
- Width: cet attribut permet d'indiquer la largeur du composant. Il n'est applicable qu'aux séparateurs horizontaux.
- Closed: l'élément doit être à l'état fermé.
- Colour: indice de couleur.

7.6.1.2 Le composant cadre

Description

Un cadre se compose de quatre traits pleins bien visibles qui délimitent une fenêtre rectangulaire de la boîte de dialogue.

Comportement

Le composant cadre doit être inaccessible.

Fonctions d'interactivité

- Néant.

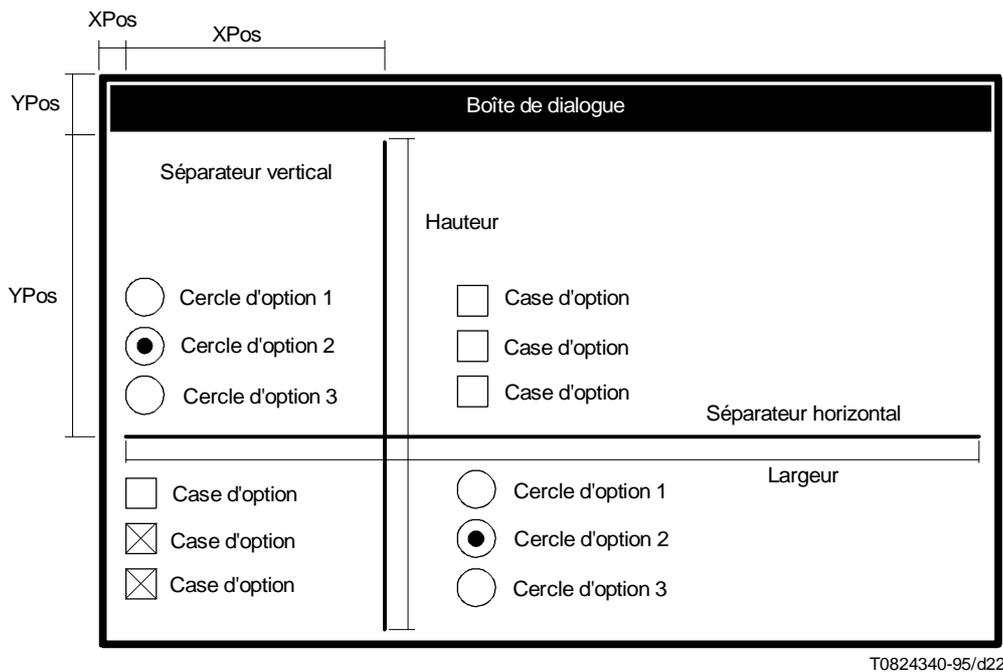


FIGURE 22/T.107
Séparateur

Aspect visuel (voir la Figure 23)

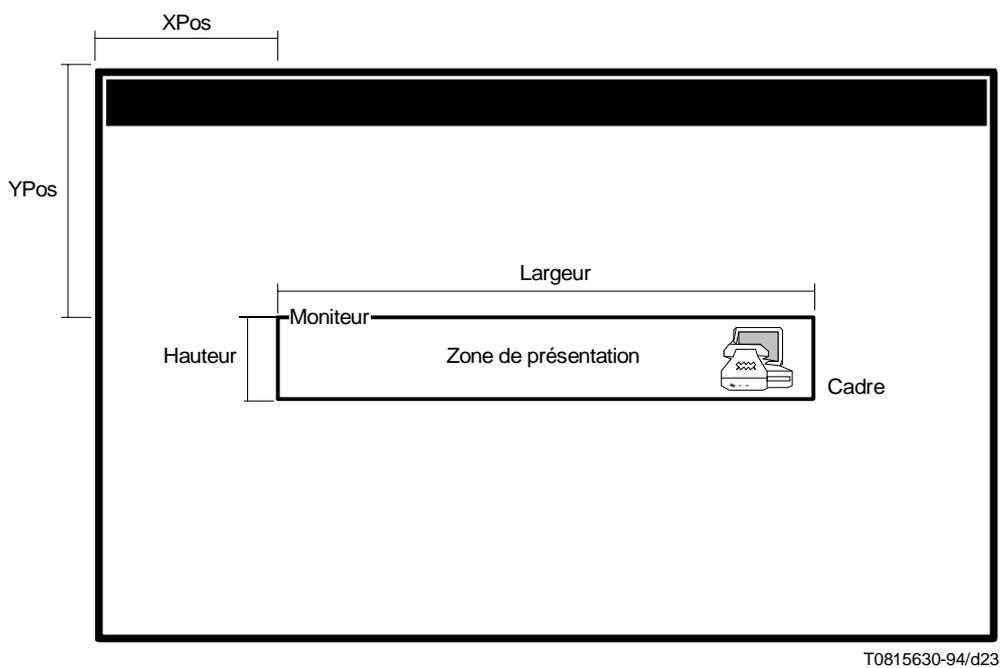


FIGURE 23/T.107
Cadre intitulé zone de présentation de texte

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du composant.
- Height: cet attribut permet d'indiquer la hauteur du composant.
- Closed: l'élément doit être à l'état fermé.
- Colour: indice de couleur.

7.6.1.3 Le composant zone de présentation de texte

Description

Le composant zone de présentation de texte est une fenêtre destinée à présenter des données en mode texte à l'utilisateur. Pour l'essentiel, cet élément sert à présenter un texte qui entre dans une fenêtre (espace délimité par un composant zone de texte) ou à présenter un long texte dans lequel l'utilisateur devra naviguer à l'aide d'outils de défilement.

Le point de départ de la zone de présentation de texte se situe dans le coin supérieur gauche du rectangle délimité par le composant et le texte est affiché rangée par rangée de gauche à droite. Le texte peut contenir des attributs «dans le texte» assurant la commutation entre les polices des différents attributs et permettant de définir des «chaînes de texte sensible». Plusieurs attributs «dans le texte» peuvent apparaître, dans n'importe quel ordre; ils sont évalués, pendant l'opération d'affichage, de manière séquentielle et d'après les informations structurelles contenues dans les composants texte (attributs «élément précédent», «élément suivant»).

Défilement, bordure, titre

Si l'espace nécessaire pour la présentation des données est plus grand que l'espace attribué pour le composant, des outils de défilement vertical doivent être prévus par l'application du terminal. Le terminal peut présenter:

- des outils de défilement à côté de l'extrémité droite de la zone;
- des outils de défilement réduits sous forme de boutons recouvrant la zone;
- des touches ascenseur assurant les fonctions de défilement.

L'espace nécessaire pour les outils de défilement est compris dans les dimensions globales du composant.

Le composant peut être enrichi d'une bordure. Si elle est présente, cette bordure doit être incluse dans les dimensions globales du composant. Si une bordure est demandée, un cadre doit être dessiné par le terminal. Le dessin de ce cadre dépend du terminal. La largeur du cadre ne doit pas dépasser 1/160 NDC. Si des outils de défilement figurent à côté de l'extrémité droite du composant (à l'extérieur de celui-ci) il faut élargir le cadre en déplaçant sa bordure pour inclure ces outils visuellement dans la boîte de dialogue.

Opération d'agrandissement plein écran

La taille de la boîte de dialogue peut être modifiée par l'utilisateur à l'aide des outils de défilement fournis par l'interface utilisateur graphique (GUI) (*graphical user interface*) locale et d'un bouton spécifique d'agrandissement plein écran. La stratégie recommandée est décrite au 7.1.5, où sont formulées des recommandations par défaut. Si l'attribut «NoFormat» est sélectionné pour le composant, les règles suivantes s'appliquent:

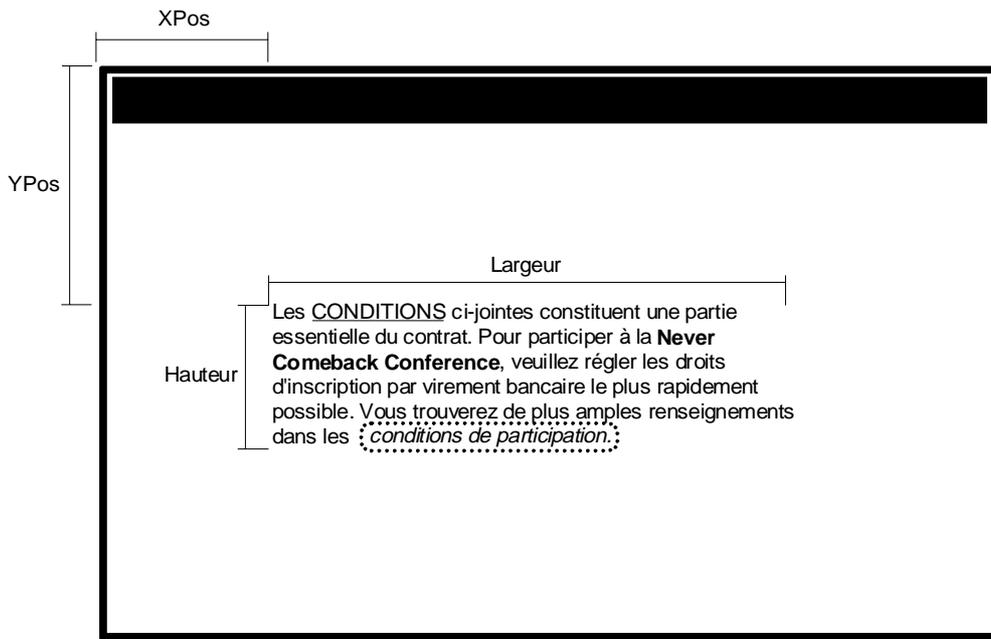
- un élargissement vertical ne doit pas modifier la forme du texte ni nécessiter l'adjonction d'un texte supplémentaire;
- un rétrécissement vertical doit couper le texte au ras de la nouvelle bordure;
- un élargissement horizontal doit ajouter de nouvelles lignes de texte;
- un rétrécissement horizontal doit couper le texte.

Fonctions d'interactivité

- Activation et validation du texte sensible, définies à l'aide d'attributs «dans le texte».

NOTE – Pour transférer une quantité importante de données en mode texte, il est recommandé d'utiliser, non pas la définition du mode texte direct, mais l'objet Text Resource et de transmettre les données au moyen de la commande de transfert de ressources VEMMI.

Aspect visuel (voir la Figure 24)



T0824350-95/d24

FIGURE 24/T.107

Zone de présentation de texte

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du composant.
- Height: cet attribut permet d'indiquer la hauteur du composant.
- NoScrollingTools: le terminal ne doit pas dessiner d'outils de défilement.
- Format: il convient d'appliquer, pour l'agrandissement plein écran de la boîte de dialogue, les quelques règles de formatage décrites ci-dessus.
- NoBorder: l'élément ne doit pas être encadré d'une bordure.
- InitialFnt: FIN de la police d'attributs à appliquer au premier texte.

- Autoscroll: une modification d'un composant auquel fait référence la zone de texte provoque l'affichage immédiat de ce composant modifié.
- InText: contenu en mode texte pouvant inclure des attributs dans le texte et des références à des objets de police.
- TextCompRef: référence à un composant texte.
- Closed: l'élément doit être à l'état fermé.

7.6.1.4 Le composant texte

Description

Ce composant est utilisé pour diviser de longues données de texte en unités (composants texte) et pour définir l'information structurelle nécessaire (concaténation de composants texte) pour l'affichage dans une zone de texte. Cette information structurelle est constituée de références aux composants texte précédent et suivant, définis par les attributs «NextText» et «PreviousText».

Concept affichage

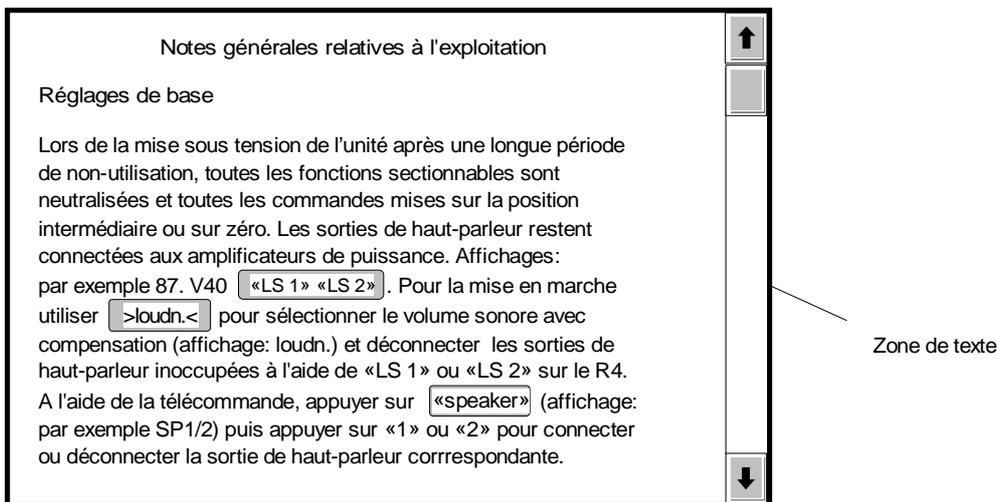
Les composants ne sont pas nécessairement de la même longueur. Pendant l'affichage de composants texte concaténés, l'utilisateur doit avoir l'impression d'un texte continu. La continuité ne doit pas être interrompue par une commutation d'un composant à un autre. Chacun des composants successifs doit offrir des possibilités de défilement vertical, rangée par rangée et page par page. Cette continuité d'affichage est interrompue dans deux cas:

- un composant n'est concaténé à rien d'autre; pas d'attribut «NextText» pendant un défilement avant ni d'attribut «PreviousText» pendant un défilement arrière. L'affichage ne continue pas parce que le texte qui vient juste d'être affiché relève d'un sujet qui a atteint sa fin logique. La Figure 29 montre deux séquences de composants concaténés: CIN 7, 8, 9 et CIN 17;
- «sauts» dus à la validation d'un texte sensible faisant partie des composants affichés ou d'autres composants interactifs de la boîte de dialogue. L'attribut texte sensible peut contenir comme action locale une commande élémentaire spécifique, par exemple «ouverture d'un composant de l'objet de rattachement». Le composant de rattachement désigné est un composant texte qui sera affiché sur la position de départ et non pas à la suite du dernier composant affiché. Cet effet est représenté sur les Figures 27 et 28.

Le texte à afficher est contenu dans la définition du composant, dans un autre composant texte désigné ou dans un objet ressource texte désigné. Si un tel objet ressource n'est pas disponible dans le terminal, il peut être demandé à l'application assurée par le serveur à l'aide de la commande VEMMI_Object_Retransmission et la règle B doit être appliquée (voir 5.3.2). Une telle demande peut être lancée par un défilement avant jusqu'à des données qui ne figurent pas encore dans le terminal, mais en général il n'existe pas de relation directe entre l'action de défilement et la demande de retransmission d'un objet, le terminal conservant les objets déjà reçus.

NOTE – Pour transférer une quantité importante de données en mode texte, il est recommandé d'utiliser, non pas la définition du mode texte direct, mais l'objet Text Resource et de transmettre les données au moyen de la commande de transfert de ressources VEMMI.

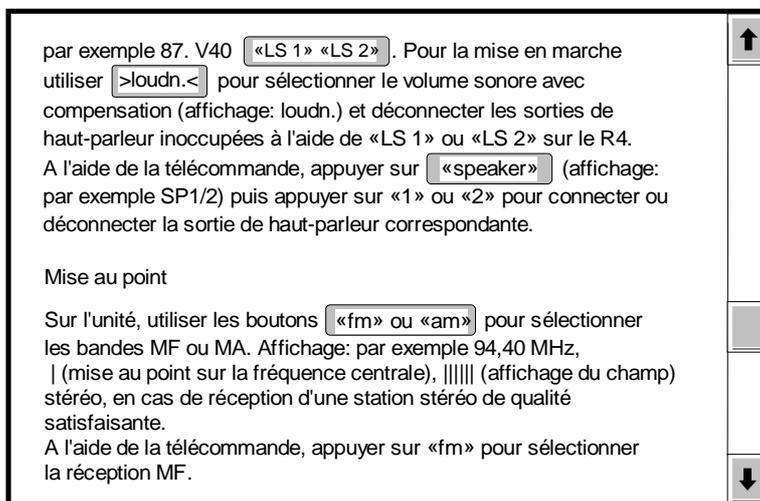
Aspect visuel (voir les Figures 25 à 28)



T0823990-95/d25

FIGURE 25/T.107

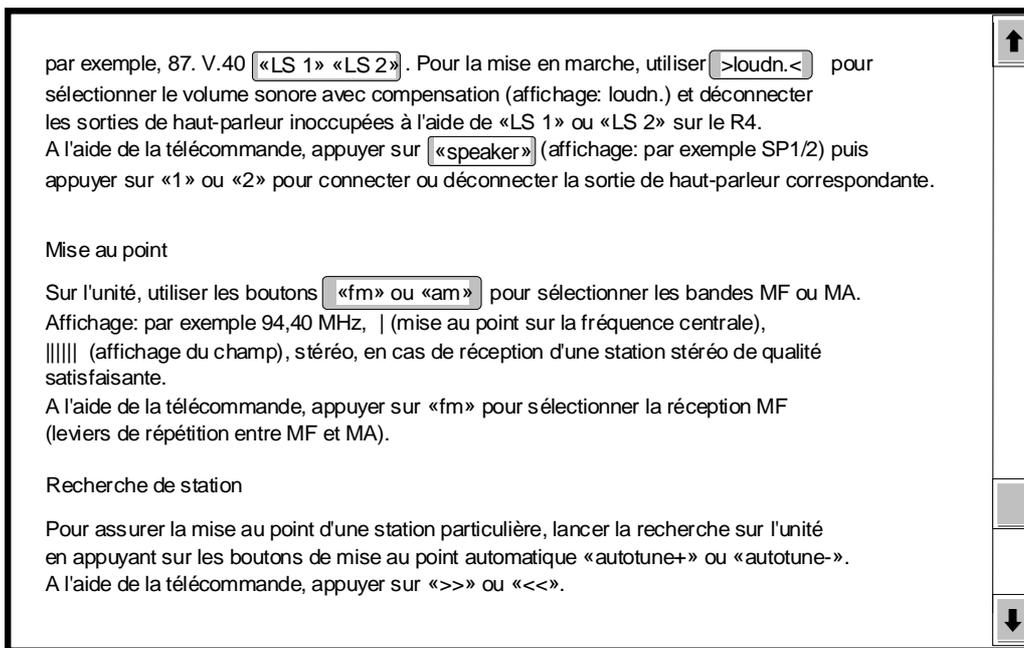
Zone de texte avec composant(s) texte, barres et défilement et textes sensibles (l'attribut «NoFormat» n'est pas sélectionné). L'utilisateur lance le programme et fait défiler l'image vers le bas...



T0823560-95/d26

FIGURE 26/T.107

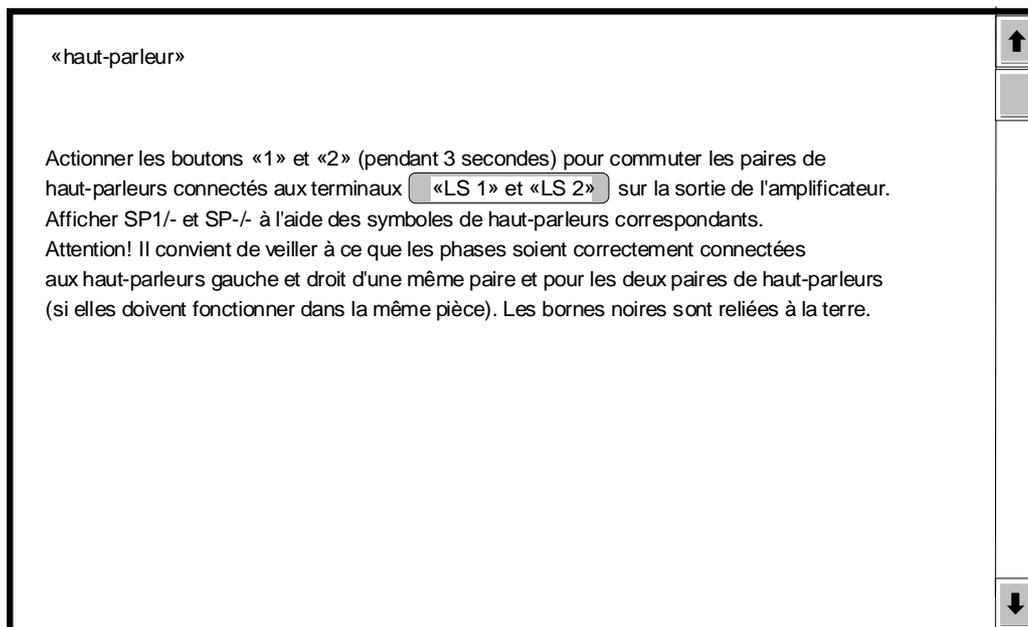
... puis agrandit la fenêtre dans les deux sens ...



T0823570-95/d27

FIGURE 27/T.107

... et enfin, clique sur le petit groupe d'instructions «speaker»



T0823580-95/d28

FIGURE 28/T.107

Le texte du paragraphe «haut-parleur» est affiché

L'architecture des composants pour la séquence d'affichage représentée aux Figures 25 à 29 peut être la suivante (les attributs dans le texte ne sont pas énumérés):

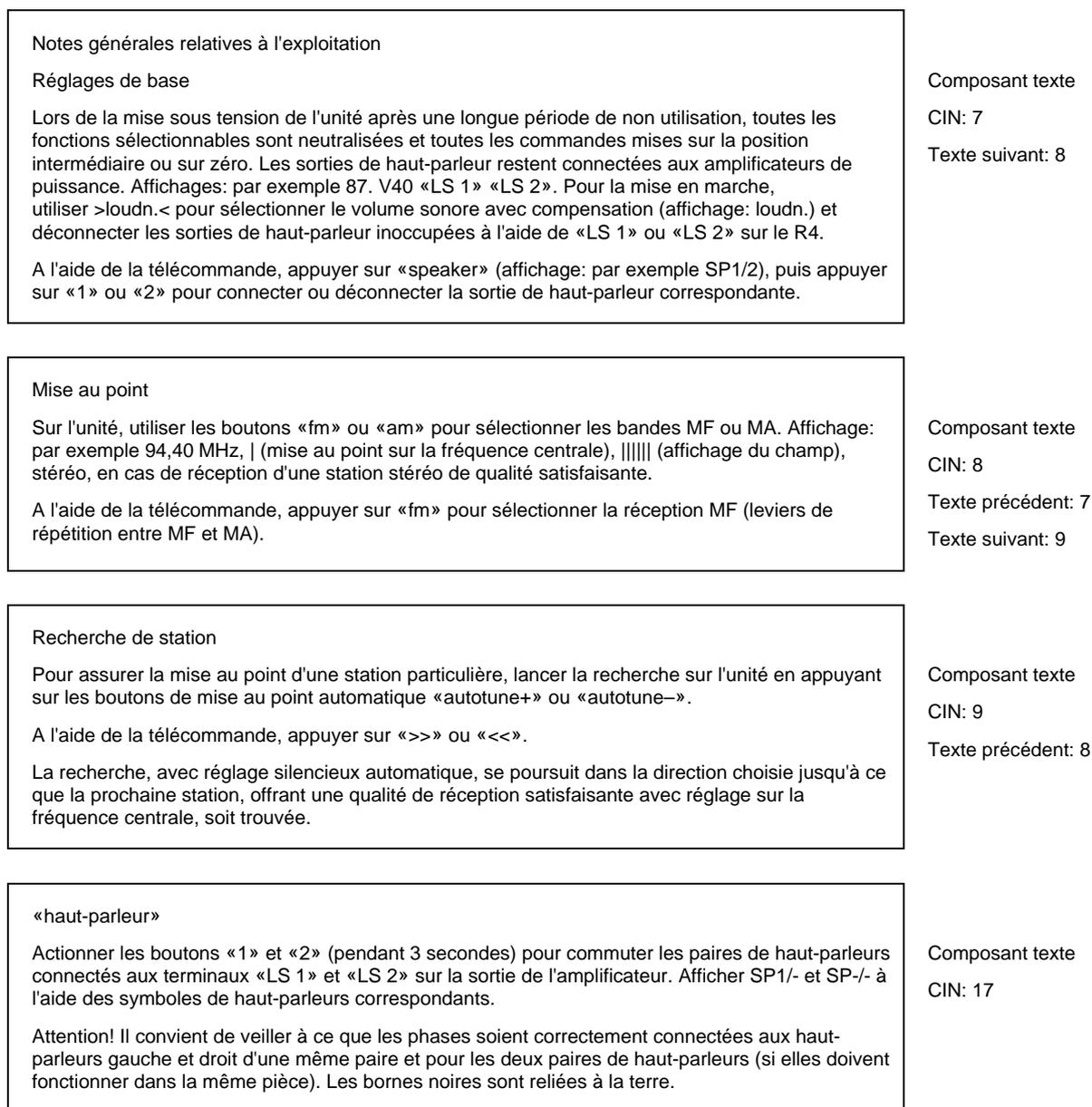


FIGURE 29/T.107

Architecture des composants texte

L'interactivité locale peut être mise en œuvre par une coordination appropriée des actions locales (paramètre LocActVal) faisant partie du texte sensible et des autres éléments interactifs de la boîte de dialogue. Dans le cas pris en exemple, l'utilisateur peut sauter du texte de la Figure 27 à celui de la Figure 25 au moyen du texte sensible «Speaker», si ce texte sensible contient l'action locale «ouverture du composant 7».

Attributs

- CIN: numéro d'identification de composant.
- PreviousText: CIN du composant texte concaténé amont.
- CurrentText: définition du texte, y compris les références des objets ressource de police, des objets ressource texte et des composants texte sensible.
- NextText: CIN du composant texte concaténé aval.

7.6.1.5 Le composant texte sensible

Ce composant définit les opérations d'activation et de validation des chaînes de texte sensible. Celles-ci font partie des composants texte.

Aspect visuel

Voir 7.6.1.4.

Attributs

- CIN: numéro d'identification de composant.
- NotAccessible: l'élément n'est pas accessible.
- LocActAct: cet attribut spécifie les actions locales associées au composant et déclenchées par son activation.
- LocActVal: cet attribut spécifie les actions locales associées au composant et déclenchées par sa validation.

7.6.1.6 Le composant zone d'affichage de graphiques

Description

La zone d'affichage de graphiques est un rectangle destiné à l'affichage de données graphiques (phototrame, données vidéotex, etc.) Une phototrame est identifiée par un BIN. Il existe trois modes d'affichage différents: la phototrame peut être positionnée au centre de la zone d'affichage de graphiques; elle peut être étirée pour couvrir toute la zone ou être présentée en juxtaposé. Dans tous les cas, la phototrame ne doit pas déborder de l'espace réservé au moyen de ce composant. Si la phototrame identifiée n'existe pas pendant que le composant est ouvert, elle n'a pas d'effet visuel.

Des données vidéotex sont identifiées par un VIN. Dans tous les cas, ces données ne doivent pas déborder de l'espace réservé au moyen de ce composant. Si le contenu de données vidéotex identifié n'existe pas pendant que le composant est ouvert, il n'a pas d'effet visuel.

Comportement

Le composant zone d'affichage de graphiques est inaccessible.

Fonctions d'interactivité

- Néant.

Aspect visuel (voir la Figure 30)

Attributs

- CIN: numéro d'identification de composant.
- XPos: position horizontale du coin supérieur gauche du rectangle d'affichage en phototrame.
- YPos: position verticale de l'angle supérieur gauche du rectangle d'affichage en phototrame.
- Width: largeur du rectangle d'affichage en phototrame.
- Height: hauteur du rectangle d'affichage en phototrame.
- Closed: l'élément doit être à l'état fermé.
- DispType: spécifie le mode d'affichage de la phototrame: centré, étiré (par défaut) ou en juxtaposé.
- BIN: numéro d'identification de phototrame.
- VIN: numéro d'identification de mode vidéotex.

7.6.1.7 Le composant poussoir

Description, comportement, fonctions d'interactivité

Voir 7.4.1.1.

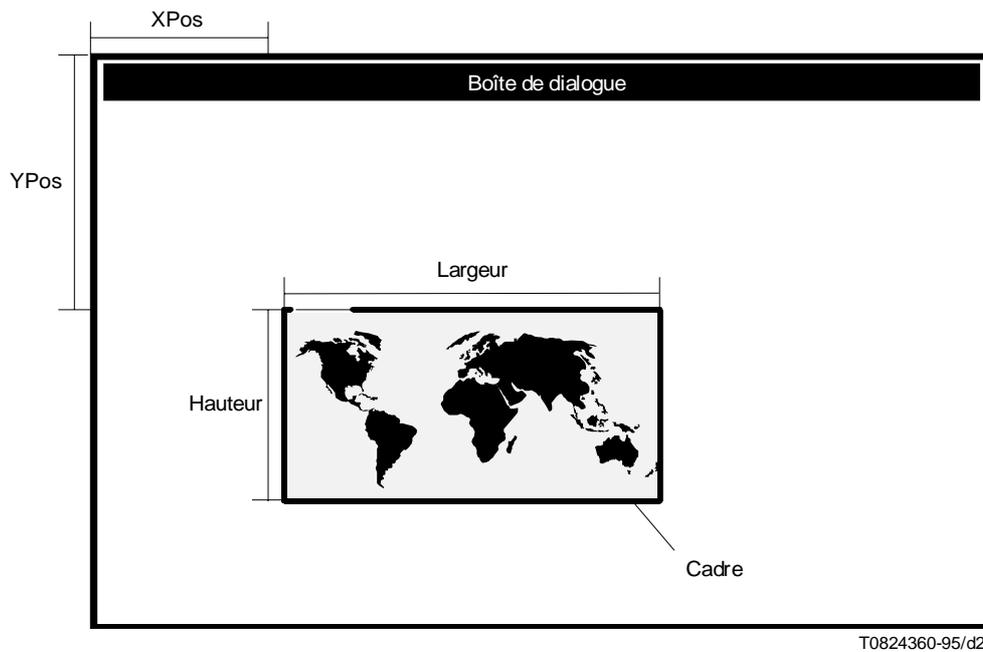


FIGURE 30/T.107
Cadre avec phototrame

Aspect visuel (voir la Figure 31)

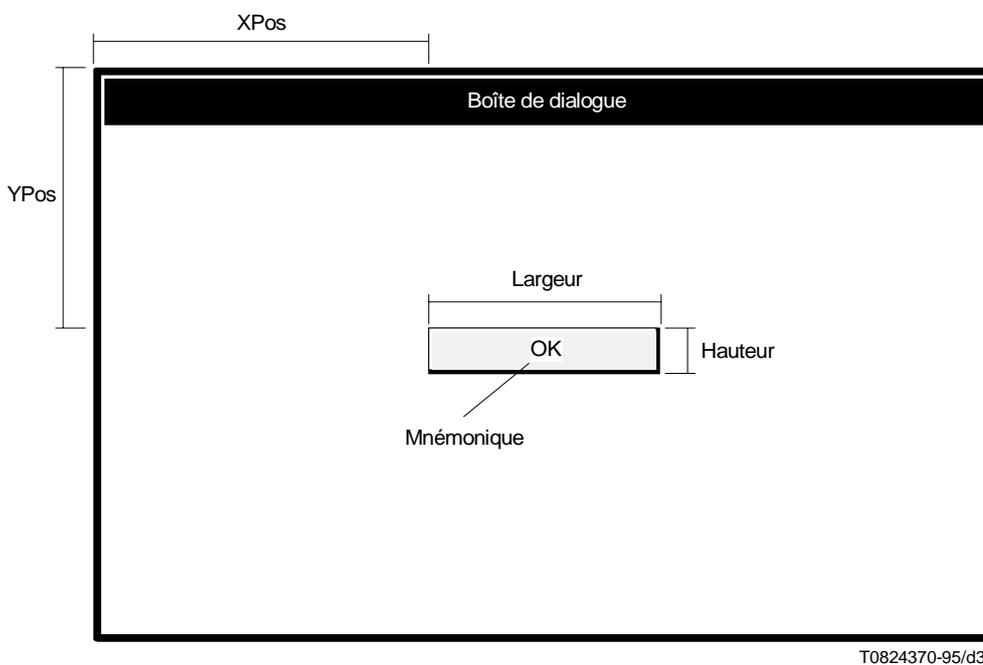


FIGURE 31/T.107
Poussoir

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du composant.
- Height: cet attribut permet d'indiquer la hauteur du composant.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.
- BIN: numéro d'identification de phototrame (BIN) du composant.
- Text: contenu en mode texte du composant.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.6.1.8 Le composant zone de saisie de texte

Description

Le composant zone de saisie de texte comprend deux items:

- l'étiquette de texte;
- la zone de saisie.

Une police d'attributs peut être associée à l'étiquette de texte. La zone de saisie commence immédiatement après l'étiquette en mode texte. Le type des données de saisie peut être spécifié par l'application. Les types prédéfinis suivants sont disponibles:

- caractères quelconques en mode texte;
- données alphabétiques (A...Z, a...z, caractères diacritiques);
- données numériques (0...9, +, -, virgule, point, espace);
- données alphanumériques (alphabétiques et numériques).

Un des attributs du composant zone de saisie de texte spécifie si les saisies de l'utilisateur sont ou non renvoyées en écho. L'application VEMMI peut définir un seul caractère pour renvoyer en écho toutes les saisies de l'utilisateur.

Une touche mnémorique permet d'activer le composant zone de saisie de texte (voir 7.1.7).

Comportement

Si l'espace attribué à la zone de saisie dans un composant zone de saisie de texte ne comportant qu'une seule ligne n'est pas suffisant pour afficher tous les caractères saisis par l'utilisateur, la zone de saisie doit offrir des possibilités de défilement horizontal. Si l'espace attribué à la zone de saisie dans un composant zone de saisie de texte comportant plusieurs lignes n'est pas suffisant pour afficher tous les caractères saisis par l'utilisateur, cette zone de saisie doit offrir des possibilités de défilement vertical. De façon à afficher tous les caractères possibles, il est recommandé d'attribuer un espace suffisant pour les saisies utilisateur. Si le nombre maximal de caractères saisis est atteint, il y a lieu que cela soit indiqué à l'utilisateur.

Contraintes

La longueur maximale de l'étiquette de texte est limitée à la première ligne dans la largeur du composant zone de saisie de texte. Pour les zones de saisie d'une seule ligne, l'étiquette est placée devant la zone de saisie et est alignée à gauche, la zone de saisie étant alignée à droite. Pour les zones de saisie comportant plusieurs lignes, l'étiquette est placée sur la première ligne et est alignée à gauche. Aucun espace n'est réservé à l'étiquette de texte si l'attribut TextLabel n'est pas présent.

Fonctions d'interactivité

- Activation.
- Défilement.
- Fonctions d'édition locale de la zone de saisie (dépendant du terminal).

Aspect visuel (voir la Figure 32)

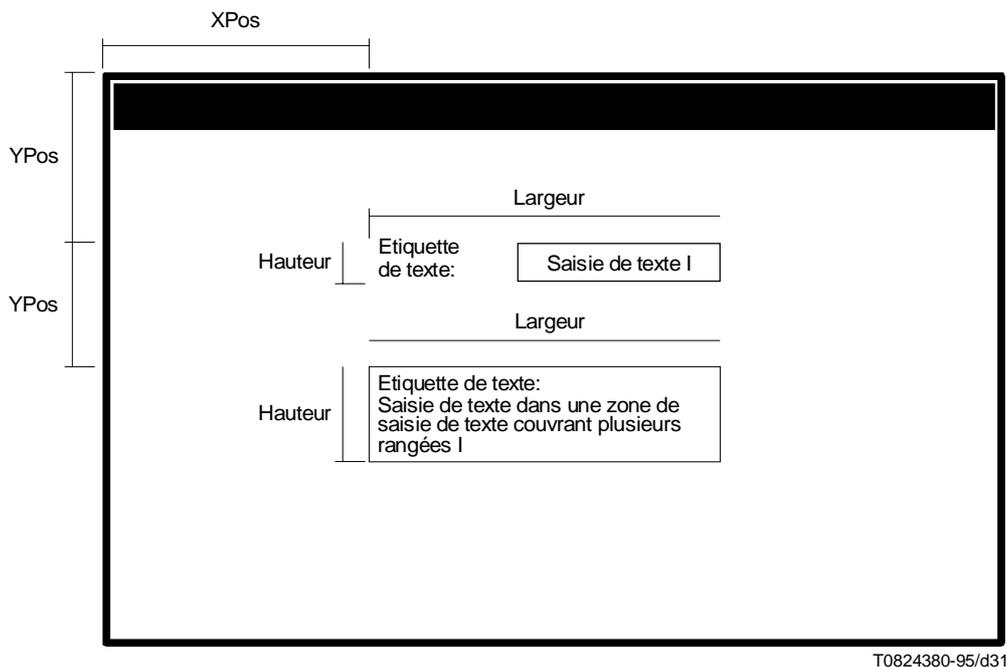


FIGURE 32/T.107
Zone de saisie de texte

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du composant.
- Height: cet attribut permet d'indiquer la hauteur du composant.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.
- MaximTxt: la commande locale plein écran peut agrandir le texte du composant proportionnellement.
- DefValue: cet attribut permet d'indiquer la valeur par défaut du composant.
- TextLabel: cet attribut permet d'indiquer l'étiquette de texte du composant.
- LabelFont: cet attribut permet d'indiquer le FIN de l'étiquette de texte.
- InputType: cet attribut permet d'indiquer les types de saisies utilisateur que le composant doit accepter.
- Echo: cet attribut permet d'indiquer le type de renvoi en écho des saisies utilisateur.
- EchoChar: cet attribut permet d'indiquer le caractère qui doit être affiché pour renvoyer en écho des saisies utilisateur.

- **MaxChar:** cet attribut permet d'indiquer le nombre maximal de caractères que l'utilisateur peut saisir sur une seule ligne.
- **MaxLine:** cet attribut permet d'indiquer le nombre maximal de lignes que l'utilisateur peut saisir dans une zone de saisie comportant plusieurs lignes.
- **Multiline:** indique une zone de saisie comportant plusieurs lignes.
- **InputTransformation:** cet attribut permet d'indiquer si le terminal doit convertir les caractères saisis par l'utilisateur en caractères majuscules ou en caractères minuscules ou si aucune conversion ne doit être effectuée.
- **LocActAct:** cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- **LocActVal:** cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.6.1.9 Le composant case d'option

Description

Le composant case d'option comprend deux items:

- l'item étiquette de texte;
- l'item croix.

Une touche mnémonique permet d'activer ce composant (voir 7.1.7).

L'item croix doit permettre la prise en charge de deux options différentes, visuellement identifiables, représentant les deux valeurs possibles suivantes de la case: cochée ou non cochée. L'espace attribué au composant doit être de dimensions suffisantes pour représenter cet item croix.

Comportement

Une case d'option se comporte comme un commutateur. Lorsqu'elle est activée par l'utilisateur, sa valeur passe de non cochée à cochée. La valeur des cases d'option est indépendante de celle de tout autre composant.

Fonctions d'interactivité

- Activation.
- Commutation entre les valeurs cochée et non cochée.
- Validation.

Aspect visuel (voir la Figure 33)

Attributs

- **CIN:** cet attribut permet d'indiquer le numéro d'identification de composant.
- **XPos:** cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- **YPos:** cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- **Width:** cet attribut permet d'indiquer la largeur du composant.
- **Height:** cet attribut permet d'indiquer la hauteur du composant.
- **Closed:** l'élément doit être à l'état fermé.
- **NotAccessible:** l'élément n'est pas accessible.
- **TextLabel:** cet attribut permet d'indiquer l'étiquette de texte du composant.
- **LabelFont:** FIN de l'étiquette.
- **DefMarked:** la valeur par défaut de l'élément doit être marquée.
- **LocActAct:** cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- **LocActVal:** cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

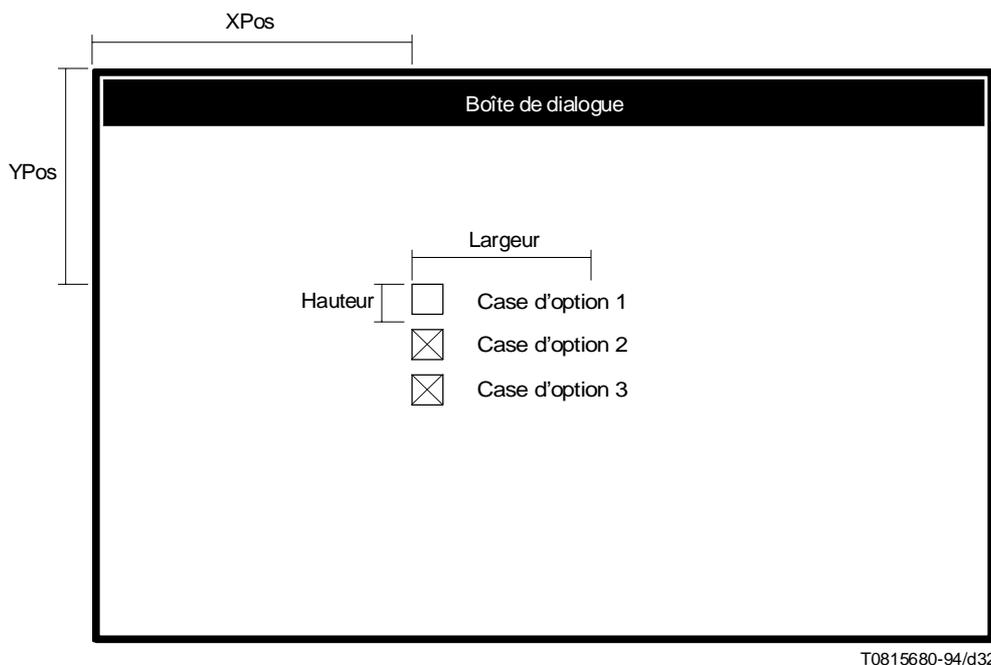


FIGURE 33/T.107
Case d'option

7.6.1.10 Le composant cercle d'option

Description

Le composant cercle d'option comprend deux items:

- l'item étiquette de texte;
- l'item croix.

Une touche mnémotique permet d'activer le composant (voir 7.1.7).

L'item croix doit permettre la prise en charge de deux options différentes, visuellement identifiables, représentant les deux valeurs possibles suivantes du cercle: cochée ou non cochée. L'espace attribué au composant doit être de dimensions suffisantes pour représenter cet item croix.

Normalement, les cercles d'option sont groupés en barrettes afin d'offrir un champ d'options exclusives.

Comportement

A un moment donné, un seul cercle d'option doit être coché dans une barrette de cercles d'option. Le cochage d'un second cercle d'option annule le cochage du cercle d'option précédemment coché, quel que puisse être son état (ouvert, fermé, accessible, inaccessible) à l'intérieur de l'application.

Fonctions d'interactivité

- Activation.
- Commutation entre les valeurs cochée et non cochée.
- Validation.

Aspect visuel (voir la Figure 34)

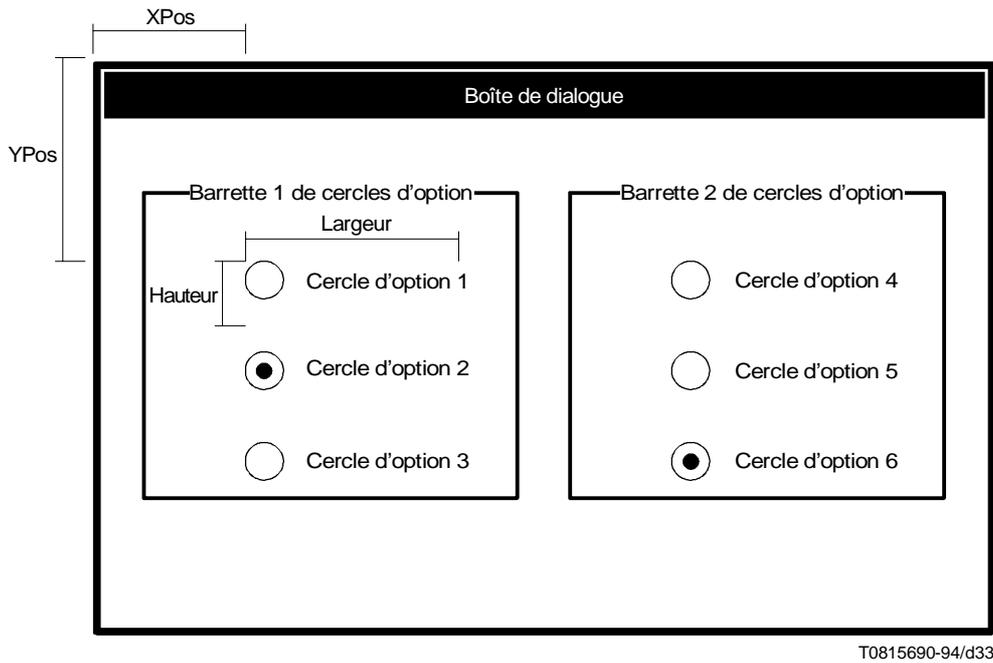


FIGURE 34/T.107

Cercles d'option avec cadre et zone de présentation de texte

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du composant.
- Height: cet attribut permet d'indiquer la hauteur du composant.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.
- TextLabel: cet attribut permet d'indiquer l'étiquette de texte du composant.
- LabelFont: FIN de l'étiquette.
- Group: cet attribut permet d'identifier la barrette de cercles d'option. Un seul cercle peut être coché à un moment donné dans une barrette.
- DefMarked: la valeur par défaut de l'élément doit être marquée.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.6.1.11 Le composant boîte d'énumération

Description

Le composant boîte d'énumération est une fenêtre rectangulaire dans laquelle une liste d'items en mode texte est présentée à l'utilisateur pour choix. Si le nombre total d'options dépasse le nombre d'items présentés dans la fenêtre, des possibilités de défilement doivent être offertes à l'utilisateur, au moyen de commandes de défilement spéciales. Si toute la liste est visible, les commandes de défilement doivent être invisibles. Dans ce cas, l'espace attribué au composant doit être de dimensions suffisantes pour représenter les dispositifs de défilement. Les outils devant assurer les fonctions de défilement (barre de défilement verticale, boutons, etc.) dépendent du terminal.

On peut ajouter des icônes aux items de liste pour en souligner visuellement la signification. On peut redimensionner la phototrame identifiée pour l'adapter à la représentation locale de la boîte d'énumération.

La liste est entièrement composée d'items de liste. Il ne doit y avoir ni retour de chariot ni changement de ligne dans le contenu en mode texte d'un item de liste. Si l'attribut `sorted` est sélectionné, les items de liste doivent être classés par ordre alphabétique. Si cet attribut n'est pas sélectionné, la liste doit être présentée dans l'ordre croissant des index d'items de liste. Toutefois, la correspondance entre les items de liste et les index est toujours maintenue.

Comportement

Lorsque le composant boîte d'énumération est actif, le terminal doit offrir à l'utilisateur des possibilités de recherche par décalage. Une boîte d'énumération contient une liste d'items que l'utilisateur peut faire défiler et sélectionner. Les choix offerts à l'utilisateur peuvent être uniques ou multiples, selon les attributs de la liste.

Des actions locales peuvent être associées à chaque item d'une boîte d'énumération. En outre, la boîte d'énumération peut aussi comporter des actions locales qui lui sont associées. Après activation et validation d'un item de boîte d'énumération, les actions locales associées à cet item doivent être exécutées. Si aucune action locale n'est associée à l'item sélectionné, l'action locale associée à la boîte d'énumération elle-même doit être exécutée. Dans ce cas, un OIN quelconque apparaissant dans des commandes élémentaires spécifiques du composant boîte d'énumération est augmenté de la valeur de l'index de l'item de la boîte d'énumération.

Une boîte d'énumération à choix multiple ne peut pas être validée.

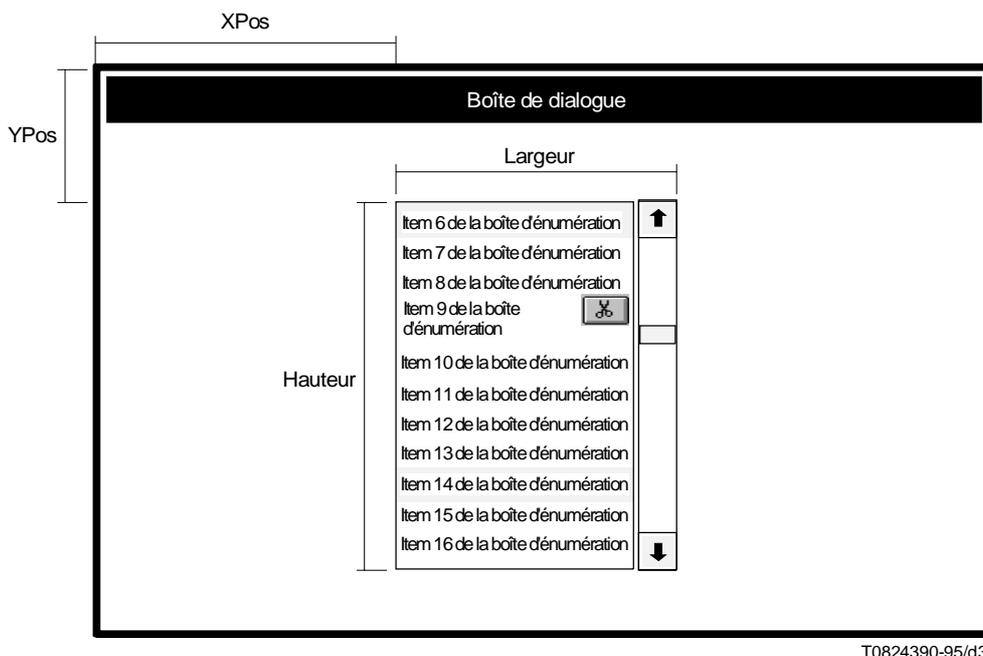
Fonctions d'interactivité

- Activation.
- Choix unique ou multiple.
- Validation (choix unique).
- Défilement.
- Décalage.

Aspect visuel (voir la Figure 35)

Attributs

- `CIN`: cet attribut permet d'indiquer le numéro d'identification de composant.
- `XPos`: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- `YPos`: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- `Width`: cet attribut permet d'indiquer la largeur du composant.
- `Height`: cet attribut permet d'indiquer la hauteur du composant.
- `Closed`: l'élément doit être à l'état fermé.
- `NotAccessible`: l'élément n'est pas accessible.
- `DefItem`: cet attribut permet d'indiquer l'index de l'item de liste qui est sélectionné par défaut lorsque le composant est ouvert pour la première fois.
- `Sorted`: les items de liste sont présentés par ordre alphabétique.
- `MultipleChoice`: l'élément admet plusieurs choix.
- `LocActAct`: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- `LocActVal`: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.



T0824390-95/d34

FIGURE 35/T.107
Boîte d'énumération

Attributs d'item de liste

- **ListIndex:** cet attribut permet d'indiquer, dans la description du composant, l'index de l'item de liste suivant. Il permet de changer, par une commande de modification, un item de liste spécifique. Les index doivent être définis par ordre croissant. Ils ne peuvent pas se suivre, pour la commodité de mise à jour de la liste (insertion d'items de liste). Le terminal doit présenter à l'utilisateur une liste d'index successifs, classés par ordre croissant, correspondant au texte de la liste. Si une commande de modification utilise un index d'item de liste qui existe déjà, l'item de liste existant est remplacé. Si une commande de modification utilise un index d'item de liste qui n'existe pas dans le terminal, le nouvel item de liste doit être ajouté. Si une commande de modification désigne un index qui existe déjà mais que dans cette commande l'attribut «ListText» correspondant manque, l'item de liste désigné dans le terminal doit être supprimé.
- **ListText:** cet attribut permet d'indiquer le contenu en mode texte d'un item de liste.
- **BIN:** BIN du composant.
- **ItemActAct:** cet attribut permet d'indiquer le code de l'action locale associée à l'item et déclenchée par son activation.
- **ItemActVal:** cet attribut permet d'indiquer le code de l'action locale associée à l'item et déclenchée par sa validation.

7.6.1.12 Le composant boîte de combinaison

Description

Le composant boîte de combinaison comprend deux parties:

- une boîte d'énumération à choix unique;
- une zone de texte d'une seule ligne située en haut de la liste.

Si le nombre total d'options dépasse le nombre d'items présentables dans la fenêtre, des possibilités de défilement doivent être offertes à l'utilisateur, au moyen de commandes de défilement spéciales. Dans ce cas, l'espace attribué au composant doit être de dimensions suffisantes pour représenter les dispositifs de défilement. Les outils devant assurer les fonctions de défilement (barre de défilement verticale, boutons, etc.) dépendent du terminal.

On peut ajouter des icônes aux items de liste pour en souligner visuellement la signification. On peut redimensionner la phototrame identifiée pour l'adapter à la représentation locale de la boîte de combinaison.

La liste est entièrement composée d'items de liste. Il ne doit y avoir ni retour de chariot ni changement de ligne dans le contenu en mode texte d'un item de liste. Si l'attribut `sorted` est sélectionné, les items de liste doivent être classés par ordre alphabétique. Si cet attribut n'est pas sélectionné, la liste doit être présentée dans l'ordre croissant des index d'items de liste. Toutefois, la correspondance entre les items de liste et les index est toujours maintenue.

En variante, une boîte de combinaison peut être déroulante: elle se compose alors d'une boîte de combinaison et d'un poussoir. Seuls la zone de texte et le poussoir sont affichés jusqu'à ce que l'utilisateur valide le poussoir associé. Cette validation provoque l'affichage de la boîte d'énumération associée.

Comportement

Lorsque le composant boîte d'énumération est actif, le terminal doit offrir à l'utilisateur des possibilités de recherche par décalage. Une boîte de combinaison contient une liste d'items que l'utilisateur peut faire défiler et sélectionner afin de compléter la zone de texte. Un paramètre du composant boîte de combinaison spécifie si la zone de texte est retraitable ou non. Si elle l'est, l'utilisateur peut saisir du texte directement dans la zone de texte. Un paramètre du composant spécifie si le texte saisi par l'utilisateur doit correspondre à un des items contenus dans la liste.

Des actions locales peuvent être associées à chaque item d'une boîte d'énumération. En outre, la boîte de combinaison peut aussi comporter des actions locales qui lui sont associées. Après activation et validation d'un item de boîte d'énumération, les actions locales associées à cet item doivent être exécutées. Si aucune action locale n'est associée à l'item sélectionné, l'action locale associée à la boîte de combinaison elle-même doit être exécutée. Dans ce cas, un OIN quelconque apparaissant dans des commandes élémentaires spécifiques du composant boîte d'énumération est augmenté de la valeur de l'index de l'item de la boîte d'énumération.

Si la boîte de combinaison est de forme déroulante, la validation du poussoir déclenche la commutation entre l'affichage et l'occultation de la boîte d'énumération.

Fonctions d'interactivité

- Activation.
- Choix unique.
- Validation.
- Défilement.
- Edition.

Aspect visuel (voir la Figure 36)

Attributs

- `CIN`: cet attribut permet d'indiquer le numéro d'identification de composant.
- `XPos`: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- `YPos`: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- `Width`: cet attribut permet d'indiquer la largeur du composant.
- `Height`: cet attribut permet d'indiquer la hauteur du composant.
- `Closed`: l'élément doit être à l'état fermé.
- `NotAccessible`: l'élément n'est pas accessible.

- DefItem: cet attribut permet d'indiquer l'index de l'item de liste qui est sélectionné par défaut lorsque le composant est ouvert pour la première fois.
- NoEdit: la zone de texte du composant ne doit pas être retraitsable.
- NoConsistency: cet attribut est appliqué si l'attribut «NoEdit» n'est pas sélectionné. Il indique que le texte saisi par l'utilisateur peut être une chaîne de texte quelconque, c'est-à-dire pas nécessairement constituée des items de texte définis.
- MaxChar: cet attribut permet d'indiquer le nombre maximal de caractères que l'utilisateur peut saisir dans le composant. Il ne doit être présent que si le paramètre «EditableInput» a pour valeur true.
- NoDropDown: l'élément ne doit pas être déroulé.
- NotSorted: les items de liste ne sont pas présentés par ordre alphabétique.
- LocActAct: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- LocActVal: cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

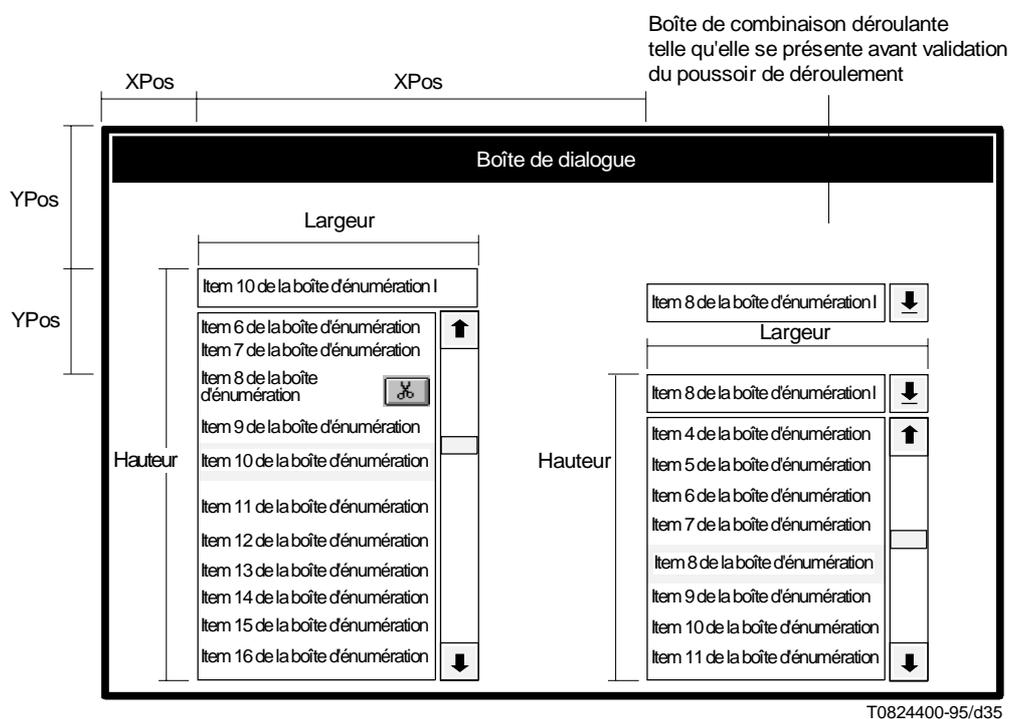


FIGURE 36/T.107
Boîte de combinaison

Attributs d'item de liste

- **ListIndex:** cet attribut permet d'indiquer, dans la description du composant, l'index de l'item de liste suivant. Il permet de changer, par une commande de modification, un item de liste spécifique. Les index doivent être définis par ordre croissant. Ils ne peuvent pas se suivre, pour la commodité de mise à jour de la liste (insertion d'items de liste). Le terminal doit présenter à l'utilisateur une liste d'index successifs, classés par ordre croissant, correspondant au texte de la liste. Si une commande de modification utilise un index d'item de liste qui existe déjà, l'item de liste existant est remplacé. Si une commande de modification utilise un index d'item de liste qui n'existe pas dans le terminal, le nouvel item de liste doit être ajouté. Si une commande de modification désigne un index qui existe déjà mais que dans cette commande l'attribut «ListText» correspondant manque, l'item de liste désigné dans le terminal doit être supprimé.
- **ListText** cet attribut permet d'indiquer le contenu en mode texte d'un item de liste.
- **BIN:** BIN du composant.
- **ItemActAct:** cet attribut permet d'indiquer le code de l'action locale associée à l'item et déclenchée par son activation.
- **ItemActVal:** cet attribut permet d'indiquer le code de l'action locale associée à l'item et déclenchée par sa validation.

7.6.1.13 Le composant ascenseur

Description

Le composant ascenseur permet à l'utilisateur de sélectionner une valeur analogique en déplaçant un pavé de circulation le long d'une barre (horizontale ou verticale) entre une valeur minimale et une valeur maximale. Les intervalles sont déterminés par l'application.

Le composant ascenseur comprend les items suivants:

- a) l'ascenseur (avec son pavé de circulation coulissant, dont la position initiale doit correspondre à la valeur par défaut, déterminée par l'application);
- b) l'étiquette de texte (désignant l'ascenseur);
- c) l'étiquette MinLabel (la valeur la plus faible qui puisse être sélectionnée), facultative;
- d) l'étiquette MaxLabel (la valeur la plus élevée qui puisse être sélectionnée), facultative;
- e) la zone DirectInputField (facultative), zone de saisie et d'affichage contenant:
 - 1) la valeur initiale au moment de l'activation de l'ascenseur;
 - 2) les valeurs en mode texte correspondant aux déplacements de l'ascenseur;
 - 3) une zone de saisie pour les saisies par l'utilisateur de la position de l'ascenseur.

Le composant ascenseur peut être orienté verticalement ou horizontalement. Tous les items doivent entrer dans le rectangle matérialisé par les dimensions du composant.

Une action locale déclenchée par la validation doit être exécutée lorsque l'utilisateur désactive le composant.

Comportement

Lorsque le composant ascenseur est actif, on peut déplacer le pavé de circulation en appuyant sur les touches de déplacement du curseur ou en cliquant avec la souris. Si la zone DirectInputField facultative est présente, l'utilisateur peut introduire des valeurs directement en manipulant les touches. Les valeurs modifiées doivent s'afficher immédiatement dans la zone DirectInputField. Il convient de déplacer le pavé de circulation pour le mettre en face de la valeur ainsi introduite. Autrement, lorsqu'on déplacera le pavé de circulation, la valeur affichée se rapportera aux déplacements effectués. Les étiquettes MinLabel et MaxLabel (si elles sont présentes) indiquent la fourchette des valeurs disponibles qui peuvent être sélectionnées.

Fonctions d'interactivité

- Fonction de saisie locale de la zone DirectInputField (dépendant de l'application).
- Positionnement local du pavé de circulation ascenseur.
- Validation par désactivation du composant.
- Validation.

Aspect visuel (voir la Figure 37)

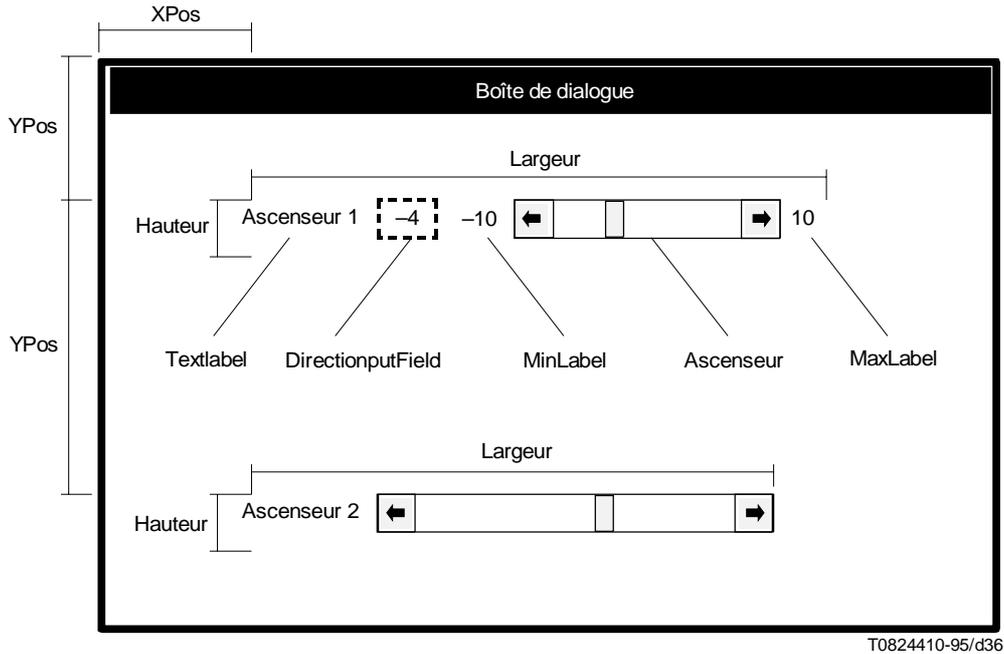


FIGURE 37/T.107
Composant ascenseur

Attributs

- CIN: cet attribut permet d'indiquer le numéro d'identification de composant.
- XPos: cet attribut permet d'indiquer – en NDC – la position horizontale de l'élément.
- YPos: cet attribut permet d'indiquer – en NDC – la position verticale de l'élément.
- Width: cet attribut permet d'indiquer la largeur du composant.
- Height: cet attribut permet d'indiquer la hauteur du composant.
- Closed: l'élément doit être à l'état fermé.
- NotAccessible: l'élément n'est pas accessible.
- Label: cet attribut permet d'indiquer une chaîne de texte alphanumérique.
- LabelFont: FIN de l'étiquette.

- **MinValue:** la valeur réglable la plus faible.
- **MaxValue:** la valeur réglable la plus élevée.
- **Increment:** taille de l'intervalle entre deux valeurs réglables.
- **InitialValue:** valeur initiale, à l'ouverture du composant ascenseur.
- **Negative:** cet attribut peut être associé aux attributs MinValue, MaxValue, InitialValue; il indique que les valeurs sont négatives.
- **DirectIn:** cet attribut indique que les attributs MinValue et MaxValue doivent être affichés et que la saisie directe par l'utilisateur doit être autorisée.
- **Vertical:** l'élément doit être présenté verticalement.
- **LocActAct:** cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par son activation.
- **LocActVal:** cet attribut permet d'indiquer le code de l'action locale associée au composant et déclenchée par sa validation.

7.6.1.14 Le composant zone sensible

Description

Un composant zone sensible est une fenêtre, non matérialisée, dans laquelle l'application permet une opération de validation. Ce composant est destiné à être utilisé en association avec un composant zone d'affichage de graphiques de la boîte de dialogue. En cas d'utilisation du composant zone sensible en association avec des composants texte, les attributs in-text de la définition du texte sensible sont plus adaptés. La zone sensible est un rectangle à l'intérieur de la boîte de dialogue.

Si l'attribut localisateur est sélectionné, le terminal doit, dès validation de cet attribut, envoyer non seulement son CIN à l'application VEMMI, mais aussi les NDC du curseur, au moment de la validation, par rapport à l'origine du composant.

Comportement

Lorsque ce composant est actif, un effet peut être affiché (filet ou pointillés, par exemple).

Contraintes

Afin d'éviter des conflits, une zone sensible ne doit pas être recouverte par d'autres composants ou items qui soient sensibles à l'interaction avec l'utilisateur.

Fonctions d'interactivité

- Activation.
- Validation.

Aspect visuel (voir la Figure 38)

Attributs

- **CIN:** cet attribut permet d'indiquer le numéro d'identification de composant.
- **XPos:** cet attribut permet d'indiquer la position horizontale de la zone sensible.
- **YPos:** cet attribut permet d'indiquer la position verticale de la zone sensible.
- **Width:** cet attribut permet d'indiquer la largeur de la zone sensible.
- **Height:** cet attribut permet d'indiquer la hauteur de la zone sensible.
- **Closed:** l'élément doit être à l'état fermé.
- **NotAccessible:** l'élément n'est pas accessible.

- Locator: avec chaque commande de compte rendu doit être envoyée la position du curseur en NDC, au moment de la validation, par rapport à l'origine du composant.
- LocActAct: cet attribut permet d'indiquer les actions locales associées au composant et déclenchées par son activation.
- LocActVal: cet attribut permet d'indiquer les actions locales associées au composant et déclenchées par sa validation. Si l'attribut Locator (localisateur) est sélectionné, chaque action de compte rendu doit également contenir les NDC du curseur, au moment de la validation, par rapport à l'origine du composant.

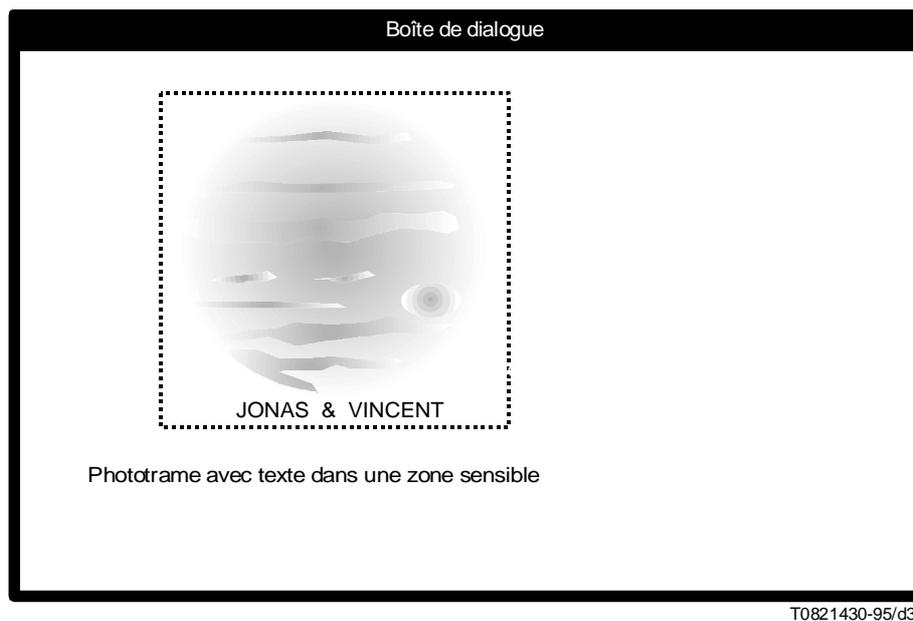


FIGURE 38/T.107

Phototrame, texte et zone sensible

7.7 La boîte de messages

Description

L'objet boîte de messages est une fenêtre rectangulaire dans la DDA qui contient des informations en mode texte.

Quatre types de messages spécifiques sont définis:

- message général;
- message d'information;
- message d'avertissement;
- message d'action.

Ces renseignements peuvent servir à ajouter une icône caractérisant le type de message en fonction de la présentation de la boîte de messages.

Un attribut de la boîte de messages spécifie s'il y a lieu que le terminal émette un son au moment de l'ouverture de cette boîte.

La boîte de messages peut être munie d'une bordure qui est normalement égale à une position de caractère. Si une bordure est requise, un cadre doit être dessiné par le terminal. Le dessin de ce cadre est fonction du terminal.

Comportement

Un attribut de la boîte de messages spécifie la durée de vie de celle-ci. Les cinq valeurs suivantes sont possibles:

- effacement par interaction quelconque avec l'utilisateur;
- fermeture par interaction quelconque avec l'utilisateur;
- effacement par validation par l'utilisateur d'un bouton défini implicitement;
- fermeture par validation par l'utilisateur d'un bouton défini implicitement;
- aucune durée de vie implicite n'est définie.

L'effacement ou la fermeture de la boîte de messages ne doit pas entraîner l'envoi d'un quelconque compte rendu au serveur.

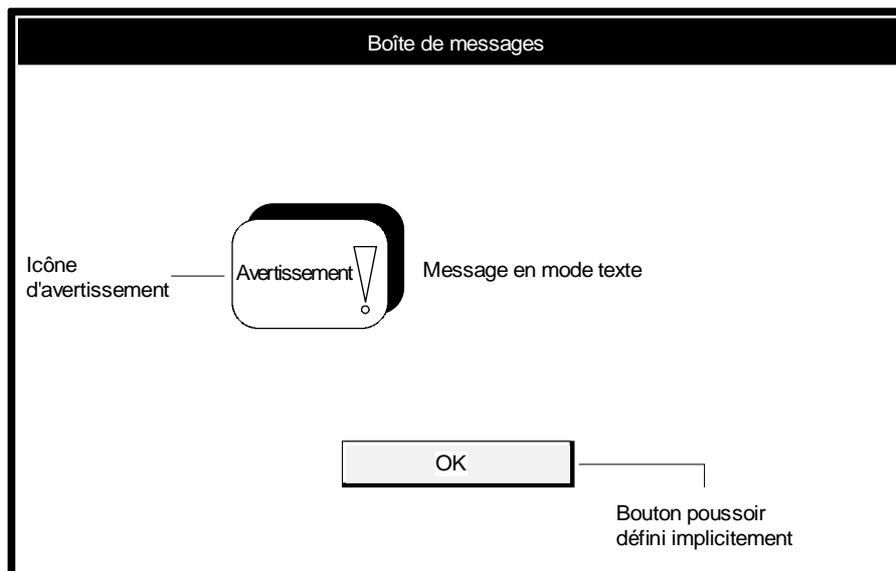
Si aucune position n'est définie par l'application VEMMI, le terminal doit centrer la boîte de messages dans la DDA.

Si aucune dimension n'est définie par l'application VEMMI, le terminal doit calculer les dimensions appropriées de la boîte de messages, de manière à avoir assez de place pour afficher le message en mode texte et l'éventuel poussoir. L'étiquette et les dimensions de cet éventuel poussoir dépendent du terminal.

Fonctions d'interactivité

- Déplacement.
- Validation.

Aspect visuel (voir la Figure 39)



T0815790-94/d38

FIGURE 39/T.107

Boîte de messages avec message d'avertissement

Attributs

- XPos: cet attribut spécifie la position horizontale de la boîte de messages.
- YPos: cet attribut spécifie la position verticale de la boîte de messages.
- Width: cet attribut spécifie la largeur de la boîte de messages.
- Height: cet attribut spécifie la hauteur de la boîte de messages.
- Closed: l'élément doit être à l'état fermé.
- MessageType: cet attribut spécifie le type du message, c'est-à-dire s'il s'agit d'un message général, d'un message d'information, d'un message d'avertissement ou d'un message d'action.
- Modal: l'élément doit être modal.
- NoBorder: l'élément ne doit pas avoir de bordure.
- AttributedText: texte du message avec attributs dans le texte (in-text).
- MaxTime: durée en secondes.
- Lifetime: événements qui conduisent à l'effacement de la boîte de messages.
- NoSound: le terminal ne doit émettre aucun son lorsque la boîte de messages est ouverte.

7.8 Objet opératif

Cet objet permet à une application de désigner un programme qui sera lié à une application VEMMI. Ce type d'objet offre une méthode pour étendre les possibilités d'une application pendant la durée d'exécution. A l'aide du service VEMMI_Create_Object l'application assurée par le serveur définit l'objet opératif. Le programme est lancé par la commande VEMMI_Open_Object. A la fin du programme, l'application reprend la direction des opérations.

Le programme proprement dit a été téléchargé par transfert de fichiers, fait partie du système d'exploitation ou sa présence a été négociée d'une autre manière. Il est désigné par son nom de fichier. Le serveur doit faire en sorte que le programme appelé soit lancé sans incident, en fournissant les paramètres appropriés pour ce programme. Une méthode souvent utilisée pour étendre les fonctions des programmes pendant la durée d'exécution est le recours à des bibliothèques de liens dynamiques.

Le programme désigné par un objet opératif peut être des types suivants:

- programme autonome;
- programme avec interface de filtrage.

Pour les objets opératifs, on utilise les services VEMMI_Create_Object, VEMMI_Open_Object et VEMMI_Destroy_Object. Le nom et les paramètres du programme sont définis comme étant des attributs de l'objet opératif.

Deux paramètres – la description du programme et l'identification de sa source – peuvent être utilisés par le terminal pour évaluer les renseignements à présenter à l'utilisateur ou simplement à enregistrer. Cette précaution est nécessaire pour aider l'utilisateur à déceler les programmes qui ont été liés à l'application VEMMI dans un dessein de sabotage et dont l'effet sur le terminal peut ne pas être anodin. Le terminal peut donner au serveur la réponse suivante: «l'utilisateur a refusé de lancer le programme!». Si les motifs de l'envoi d'une telle réponse dépendent de l'application, il n'en reste pas moins que l'information est disponible sur une base structurée. Si le terminal ne parvient pas à trouver le programme, le serveur doit en être informé par un message d'erreur spécifique.

Conditions applicables aux programmes autonomes

Le programme n'est doté d'aucune voie permettant d'échanger des données avec le gestionnaire local VEMMI pendant l'exécution. Les paramètres du programme peuvent être définis par l'application assurée par le serveur et communiqués au programme une fois celui-ci lancé. L'objet opératif n'a pas d'effet visuel sur le terminal et ne peut pas être activé directement par l'utilisateur; le programme lancé peut toutefois utiliser une interface utilisateur locale dans le cadre spécifique d'une application. Voir la Figure 40.

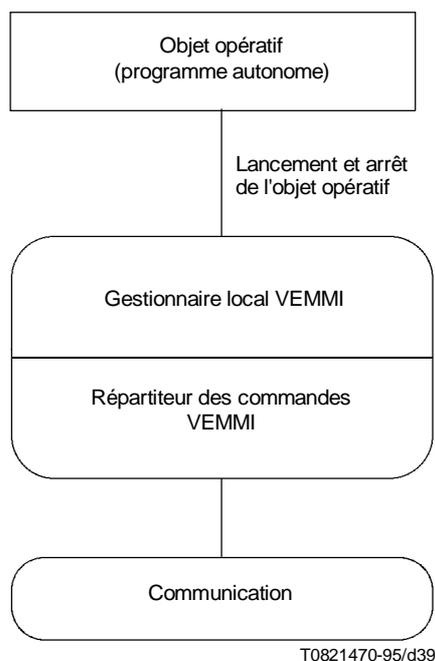


FIGURE 40/T.107
Programmes autonomes

Le répartiteur des commandes VEMMI a pour fonction de répartir ces commandes entre le gestionnaire local VEMMI et le sous-système de communication. Toutes ces interfaces dépendent du terminal et de l'application. Elles ont pour fonction, dans la présente description, d'expliquer l'intégration des deux types d'objets opératifs dans l'architecture d'un terminal VEMMI.

Conditions applicables aux programmes avec interface de filtrage

Le programme avec interface de filtrage a la possibilité d'accéder au répartiteur des commandes VEMMI. Ce répartiteur transmet les commandes à l'objet opératif, qui décide de les traiter ou de les retransmettre au gestionnaire local VEMMI. Ce mécanisme fonctionne également à l'émission. L'objet opératif n'a pas d'effet visuel sur le terminal et ne peut être activé directement par l'utilisateur; le programme lancé peut toutefois utiliser l'interface utilisateur locale dans le cadre spécifique d'une application ou par l'intermédiaire du gestionnaire local VEMMI. Les interfaces d'un tel programme, qui dépendent toutes du terminal et de l'application, ne sont pas décrites plus en détail dans la présente Recommandation. Si un objet opératif reçoit en provenance du serveur un élément de service VEMMI_Close_Object assorti de son propre OIN ou si le gestionnaire local VEMMI stoppe cet objet opératif, celui-ci cesse de fonctionner et le gestionnaire local VEMMI prend le relais. Toutes les voies d'échange de données à destination du gestionnaire local VEMMI sont désactivées. Voir la Figure 41.

Attributs

- Closed: l'élément doit être à l'état fermé.
- ProgName: nom du programme.
- ProgFilename: nom de fichier du programme.
- ProgDescr: description du programme.
- ProgAbout: identification de l'origine du programme.
- ProgPar: liste des paramètres de lancement du programme (facultative).
- ProgType: type de programme, autonome ou avec interface de filtrage.

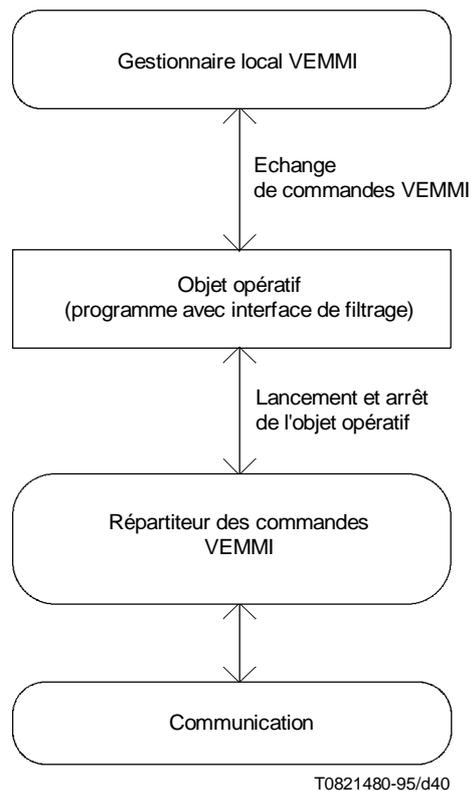


FIGURE 41/T.107

Programmes avec interfaces de filtrage

7.9 Objet de ressource en phototrame

Description

Un objet de ressource en phototrame contient la définition de la phototrame proprement dite ou simplement un rappel du fichier où figure la définition de la phototrame.

Définition de la phototrame directe

La matrice de pixels contient des indices dans la table de couleurs ou les composantes RGB de chaque pixel. La liste des attributs comprend deux parties: la première est commune aux deux cas et la seconde dépend de la définition des couleurs. La première partie se présente comme suit:

- BmWidth: largeur de la matrice en phototrame, en pixels.
- BmHeight: hauteur de la matrice en phototrame, en pixels.
- BmCompr: type de compression pour la liste de couleurs. Cette valeur n'étant pas définie pour l'instant, seules des listes de couleurs non compressées peuvent être spécifiées (facultatif).

Dans le cas de la définition d'un indice de couleur, la deuxième partie de la liste des attributs se présente comme suit:

- BmBitsPerPixel: nombre de bits par pixel. Ce nombre doit être de 1, 4 ou de 8 (facultatif, valeur par défaut: 1).
- BmClrEntry: indice de la table de couleurs sur lequel sont basés tous les autres indices de couleur répertoriés (facultatif, valeur par défaut: 0).

- **BmClrList:** liste des indices de couleur. Cette liste contient les indices de couleur $BmWidth * BmHeight$. Selon le nombre de bits par pixel, chaque indice de couleur se situe dans la gamme $0 .. 2^{BmBitsPerPixel}-1$. Les indices de couleur sont classés rangée par rangée de gauche à droite. Le point de départ est le coin inférieur gauche de la matrice en phototrame. Un indice de couleur de 0 correspond à un indice de table de couleurs égal à **BmClrEntry**. Tous les autres indices se rapportent à **BmClrEntry**.

Dans le cas de la définition d'une composante chromatique, la deuxième partie de la liste des paramètres se présente comme suit:

- **BmBitsPerComp:** nombre de bits par composante chromatique. Ce nombre doit se situer dans une fourchette de 1 à 8 (facultatif, valeur par défaut: 8).
- **BmClrCompList:** liste de triplets RGB. Cette liste contient les triplets $BmWidth * BmHeight$. Les triplets RGB sont classés rangée par rangée de gauche à droite. Le point de départ est le coin inférieur gauche de la matrice en phototrame.

Définition de la phototrame avec fichier

Ce type de phototrame est défini d'après un fichier d'images en mémoire dans le terminal. Le fichier a été transmis à l'aide d'un service de transfert de ressources ou par d'autres moyens. Un paramètre indique le codage du fichier. Si le fichier ne peut être converti en une phototrame appropriée (à la réception de l'élément de service **VEMMI_Create_Object**), le terminal doit envoyer un message d'erreur au serveur.

- **Filename:** nom du fichier où figure la définition de la phototrame.
- **PictFileType:** indice du fichier d'images (JPEG, GIF, BMP).

Les fichiers d'images sont conformes à une des spécifications suivantes:

- **JPEG:** système de base de ISO/CEI IS 10918 | Rec. UIT-T T.81 «Compression numérique et codage des images fixes de nature photographique».
- **GIF:** Graphics Interchange Format (sm) Version 89a. Compuserve Incorporated Columbus, Ohio, Etats-Unis. Bien que la version mentionnée ici soit la version 89a, un terminal ne doit assurer la prise en charge que des fonctions contenues dans la version 87.
- **BMP:** phototrame Microsoft Windows non tributaire du type d'unité (DIB) avec RLE4, RLE8 et 1, 4, 8 ou 24 bits par pixel.

7.10 Objet de ressource vidéotex

Description

Un objet de ressource vidéotex renferme le contenu en mode vidéotex proprement dit ou seulement l'indication du fichier où figure le contenu en mode vidéotex.

Attributs

- **VTX:** contenu en mode vidéotex.
- **Filename:** nom du fichier où figure le contenu en mode vidéotex.

7.11 Objet de ressource de texte

Description

Cet objet définit le contenu d'un texte comme une ressource qui peut être désignée par le «numéro d'identification de texte» (TIN). Il renferme le contenu du texte proprement dit ou l'indication du fichier où figure ce contenu. Dans les deux cas, le texte peut contenir des attributs, avec indication des objets de ressource de police.

Attributs – Définition d'une ressource de texte directe

- **Text:** texte.
- **FIN:** indication d'un objet de ressource de police.

Attributs – Définition d'une ressource de texte avec fichier

- **Filename:** nom du fichier où figure le contenu en mode texte.
- **TextFileType:** attribut d'un des types suivants:
 - texte;
 - indication des objets et du texte d'une ressource de police.

7.12 Objet de ressource de police

Cet objet combine un ensemble d'attributs de texte dans une ressource de police qui peut être désignée par le numéro d'identification de police (FIN).

Attributs

- **FnFamily:** famille de police: SWISS, ROMAN ou FIXFONT (facultatif, valeur par défaut: SWISS).
- **FnHeight:** hauteur des caractères (facultatif, valeur par défaut: 10).
- **FnBold:** caractère gras (facultatif).
- **FnUnderline:** souligné (facultatif).
- **FnItalic:** italique (facultatif).
- **FnColour:** couleur de police: indice de couleur (facultatif, valeur par défaut: noir).

7.13 Objet de métacode

L'objet de métacode contient les commandes VEMMI. Cet objet est commode pour éviter toute phase inutile de dialogue avec l'application assurée par le serveur. La commande VEMMI_Open_Object commence à traiter l'objet de métacode qui, une fois traité, passe automatique à l'état fermé. Les commandes (le contenu de l'objet de métacode) sont traitées comme si elles provenaient de l'application assurée par le serveur. Pour le contenu de l'objet de métacode, les restrictions suivantes s'appliquent:

- la création d'un objet de métacode n'est pas autorisée;
- seules les commandes assurées par le serveur sont autorisées;
- une commande VEMMI_Open_Object d'un autre objet de métacode n'est pas autorisée (une commande VEMMI_Open_Blocking_Object est autorisée).

Attributs

- **VEMMICommands:** commandes VEMMI.

7.14 Définition du type de données phototrame VEMMI

Voir 7.9.

7.15 Catalogue identifiant les types de codage de contenu VEMMI

Le Tableau 38 donne une vue d'ensemble des types de codage de contenu identifiés dans le catalogue VEMMI.

TABLEAU 38/T.107

Catalogue identifiant les types de codage de contenu VEMMI

Codage de données en mode texte:	
	Chaîne T.51 telle que définie dans l'Annexe A
	Recommandation T.52 [3]
	Texte de haute qualité VEMMI
	ISO série 8859 [13] (caractères préformés avec accent)
	ISO 10646-1 [15] (jeu de caractères multioctet (Unicode))
	Inversion de code JIS (pour caractères japonais)
Codage de données d'images fixes:	
	Annexe F/T.101 [4] (photovidéotext)
	ISO 10918-1 [16] (JPEG)
	Mode point non tributaire du type d'unité VEMMI
	Graphics Interchange Format (sm) Version 98a. CompuServe Incorporated Columbus, Ohio, Etats-Unis
	Phototrame Microsoft Windows non tributaire du type d'unité (DIB) avec RLE4, RLE8 et 1, 4, 8 ou 24 bits par pixel
Codage de données graphiques:	
	Annexes B/T.101 et Annexe C/T.101 [4] (vidéotex)
	ISO 8632 [12] (CGM)
Codage de données audio:	
	Annexe E/T.101 [4]
	Format WAVE
	Format MIDI
Codage de données d'images animées:	
	ISO 11172-2 [18] (MPEG vidéo)
	Recommandation H.261 [7] (visiophone)
Codage de données audiovisuelles:	
	ISO 11172-1 [17] (système MPEG)
	Recommandation H.320 [8] (visiophone)

8 Représentation codée complète VEMMI

8.1 Introduction

Le présent article contient la notation syntaxique, l'en-tête VEMMI et la syntaxe des commandes, des objets, des composants et des actions locales.

8.2 Notation utilisée

La notation a/b, où a, b = 0..15, indique la valeur d'un octet en numération hexadécimale.

Les bits d'un octet sont notés b₈, b₇, ..., b₁. Le bit b₈ est le bit de poids le plus fort ou le plus significatif.

La syntaxe des commandes, des objets, des composants et des actions locales est définie au moyen d'une grammaire formelle selon la notation suivante:

<symbol>	– autre que terminal;
<SYMBOL>	– terminal;
<a/b>	– terminal, valeur hexadécimale: a, b = 0..15;
<symbol>*	– 0 ou plusieurs apparitions;
<symbol>+	– 1 ou plusieurs apparitions;
<symbol>o	– facultatif (0 ou 1 apparition);
<symbol-1> := <symbol-2>	– le symbole-1 a la syntaxe du symbole-2;
<symbol-1> <symbol-2>	– le symbole-1 ou bien le symbole-2;
<symbol-1 : symbol-2>	– symbole avec indication de sa valeur;
{ observation }	– explication d'un symbole ou d'une production.

8.3 Commutation globale d'environnement de codage

L'ISO/CEI 9281 [14] décrit une technique d'identification des méthodes de codage. Le mode VEMMI pour le vidéotex fait partie des méthodes de codage relevées par l'ISO/CEI 9281 [14]. Le diagramme de la Figure 42 donne une vue d'ensemble des relations entre syntaxes de données vidéotex et environnements de codage selon l'ISO/CEI 9281 [14].

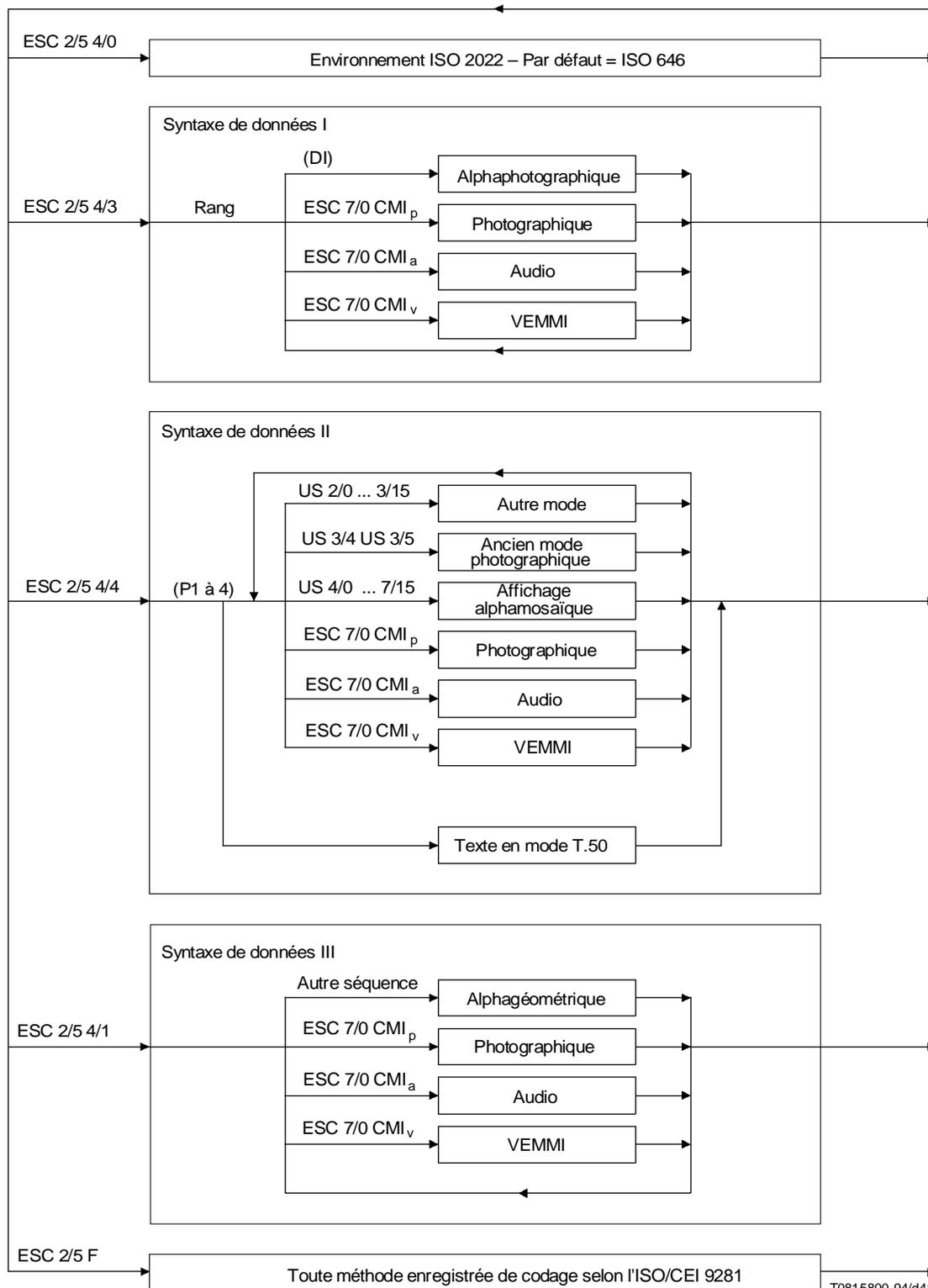
A partir d'un environnement selon l'ISO 2022 [10], on peut passer explicitement à une syntaxe de données vidéotex au moyen du code ESC 2/5 F. C'est également le mécanisme qui est utilisé pour passer d'un environnement selon l'ISO 2022 [10] à un environnement selon l'ISO/CEI 9281 [14]. Le code F («octet final») est attribué et enregistré, conformément à l'ISO 2022 [10], par l'organisme d'enregistrement selon l'ISO 2375 [11] dont l'Appendice B considère les syntaxes de données vidéotex comme des «systèmes de codage différents de ceux de l'ISO 2022 [10]». Les codes F sont 4/3 pour la syntaxe de données I du CCITT, 4/4 pour la syntaxe de données II du CCITT et 4/1 pour la syntaxe de données III du CCITT.

Etant donné qu'un terminal vidéotex commence, par défaut, à fonctionner dans une des syntaxes de données, il ne doit pas être obligatoire d'envoyer au préalable une séquence ESC 2/5 F (où F a la valeur 4/1, 4/3 ou 4/4). Le diagramme ci-dessus montre comment on peut utiliser ces codes pour passer d'une syntaxe de données à une autre avec un terminal vidéotex supportant plusieurs syntaxes de données.

8.3.1 Passage en mode VEMMI

Un terminal vidéotex opérant dans une des syntaxes de données (c'est-à-dire avec un système de codage autre que ceux qui sont décrits dans l'ISO 2022 [10]) peut passer à l'environnement du mode VEMMI selon l'ISO/CEI 9281 [14] selon les règles applicables à ces modes. Dans le cas de la commutation d'un terminal vidéotex vers l'environnement ISO/CEI 9281 [14] du mode VEMMI, on utilise le délimiteur de codage d'image (PCD) (*picture code delimiter*) du premier élément d'image (PE) (*picture element*). L'identificateur de la méthode de codage (CMI) (*coding method identifier*) sert à distinguer les méthodes de codage d'image. Dans le cas du vidéotex, il s'agira par exemple d'opérer une distinction entre données audio, photographiques et VEMMI.

NOTE – Etant donné que l'ISO/CEI 9281 [14] a été spécialement mise au point pour l'identification du codage d'image dans des environnements selon l'ISO 2022 [10], le mot «image» est souvent utilisé dans les définitions, même lorsqu'il est applicable à des données de type «audiographique» (par exemple). L'ISO a déjà accepté de faire usage de l'ISO/CEI 9281 [14] pour les informations non pictographiques.



T0815800-94/d41

- DI Spécifique de la syntaxe de données I
 - P Profil de la syntaxe de données II
 - F Code d'octet final, attribué par l'organisme d'enregistrement ISO 2022 [10]
 - CMI_a Tout identificateur de méthode de codage pour données audio du vidéotex
 - CMI_p Tout identificateur de méthode de codage pour données photographiques du vidéotex
 - CMI_v Tout identificateur de méthode de codage pour données VEMMI
- Le rang est spécifique de la syntaxe de données I

FIGURE 42/T.107
Mécanisme global de commutation d'environnement

8.3.2 Structure syntaxique ISO/CEI 9281 [14]

Le format syntaxique de haut niveau est défini dans l'ISO/CEI 9281 [14].

Dans la description de syntaxe ci-dessous, on partira de l'hypothèse du codage syntaxique à 8 éléments binaires: on utilisera donc le terme «octet», avec le bit 8 forcé à zéro. Le codage décrit dans l'ISO/CEI 9281 [14] est également valide dans un environnement à 7 éléments binaires. Dans ce dernier cas, le terme «octet» doit être interprété comme étant un «multiplet à 7 éléments binaires», le bit de poids le plus élevé (bit 8) n'étant pas utilisé.

La structure du codage est la suivante:

```

PE      ::= PCE PDE;
PCE     ::= PCD CMI LI;
PCD     ::= 01/11 07/00;
CMI     ::= PM PI;
PM      ::= 02/05 (vidéotex en mode VEMMI);
PI      ::= 04/00;
LI1)  ::= x111 1111<octet1><octet2>...<octetn>;
<octetk> ::= x10D DDDD (k = n);
          | x11D DDDD (1 = <k<n).
  
```

x indique un état indifférent.

D indique un élément binaire 0 ou 1.

Chaque élément d'information, en l'occurrence des données VEMMI codées, sera codé sous forme d'une ou de plusieurs entités d'image (PE) (*picture entity*). Chaque PE (voir la Figure 43) est formée d'une entité de commande d'image (PCE) (*picture control entity*) qui est suivie des données proprement dites, insérées dans une entité de données d'image (PDE) (*picture data entity*).

PE Entité d'image				
PCE Entité de commande d'image				PDE Entité de données d'image
PCD	CMI		LI	
Délimiteur de codage d'image	Identificateur de méthode de codage		Indicateur de longueur	
	PM Mode d'image	PI Identificateur d'image		

FIGURE 43/T.107

Structure d'une entité d'image

¹⁾ Il y a lieu de consulter 5.2.7/ISO 9281 [14] pour la description de l'utilisation de l'indicateur de longueur (LI).

NOTE 1 – En mode VEMMI pour le vidéotex, le fichier qui contient un objet VEMMI codé peut être assez grand. Un seul objet peut être transmis dans plusieurs entités d'image. L'emploi de plusieurs PE peut faciliter la bonne fin de la transmission d'un objet par l'utilisateur.

L'entité de commande d'image (PCE) est formée d'un délimiteur de codage d'image (PCD) et d'un identificateur de méthode de codage (CMI), suivis d'un indicateur de longueur (LI).

Le délimiteur de codage d'image (PCD) est une séquence fixe de deux octets: 01/11 07/00.

NOTE 2 – La séquence 01/11 correspond à ESC.

L'identificateur de méthode de codage (CMI) est formé d'un octet de mode d'image (PM), suivi d'un octet identificateur d'image (PI). Pour VEMMI, le PM est indiqué par la séquence 02/05 telle qu'enregistrée par l'ISO/CEI 9281 [14]. L'octet de PI a la valeur 04/00.

L'indicateur de longueur (LI) spécifie le nombre d'octets que comporte l'entité de données d'image. Le texte ci-dessous (tiré de ISO/CEI 9281 [14]) décrit le mode de codage de cet indicateur:

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
X	UN	ZÉRO ou UN					

Le bit b₈ de chaque octet ne doit pas être pris en compte; le bit b₇ de chaque octet doit être mis à UN. Le bit b₆ de chaque octet doit être le fanion d'extension. La valeur de LI est spécifiée en notation binaire sous la forme d'un nombre non signé au moyen des bits b₅ à b₁ ayant respectivement les poids 2⁴, 2³, 2², 2¹ et 2⁰. Si la valeur de LI est inférieure ou égale à 31, elle doit être représentée par un octet et le fanion d'extension doit être mis à ZÉRO. Si la valeur de LI est supérieure à 31, elle doit être représentée par plusieurs octets. La partie la plus significative de cette valeur doit être enregistrée dans le premier octet. Le fanion d'extension doit être mis à UN dans tous les octets sauf dans le dernier où il doit être mis à ZÉRO.

Valeurs de LI:

- 1 octet: X111 1111
- 2 octets: 0 ≤ LI ≤ 2⁵ – 1
- 3 octets: 2⁵ ≤ LI ≤ 2¹⁰ – 1
- ⋮ ⋮ ⋮ ⋮ ⋮ ⋮
- n octets: 2⁵⁽ⁿ⁻¹⁾ ≤ LI ≤ 2⁵ⁿ – 1

Exemples:

LI = 31

X	1	0	1	1	1	1	1
---	---	---	---	---	---	---	---

LI = 33

X	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---

Premier octet

X	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Deuxième octet

8.4 Syntaxe de commande VEMMI

Commande de serveur	Syntaxe de commande
VEMMI_Set_Translation_Mode	<translation mode opc> <translation mode>
VEMMI_Open	<open opc> <setup entity>
VEMMI_Close	<close opc>
VEMMI_Resume	<resume opc> <mode>
VEMMI_Suspend	<suspend opc>
VEMMI_Close_All	<close all opc>
VEMMI_Identify_Term_Cap	<term cap request opc> <cap request>
VEMMI_Set_Options	<set options opc> <text type list>o
VEMMI_Reset_Col_Table	<reset col opc>
VEMMI_User_Lock	<user lock opc>
VEMMI_User_Unlock	<user unlock opc>
VEMMI_Open_Object	<open object opc> <oin spec>
VEMMI_Close_Object	<close object opc> <oin spec>
VEMMI_Destroy_Object	<destroy object opc> <oin spec>
VEMMI_Obj_Access_Disable	<access disable opc> <oin spec>
VEMMI_Obj_Access_Enable	<access enable opc> <oin spec>
VEMMI_Create_Object	<create object opc> <entity create object>
VEMMI_Delete_Outdated_Objects	<delete outdated objects opc> <entity versionlist>+
VEMMI_Modify_Component (voir aussi 9.6)	<modify comp opc> <entity modify comp>
VEMMI_Obj_Location_Change	<object loc change opc> <oin> <xloc> <yloc>
VEMMI_Load_Col_Table	<load coltable opc> <entity load coltable>
VEMMI_Open_Application	<open appl opc> <entity open appl>
VEMMI_Store_Objects	<store objects opc> <oin spec>
VEMMI_Erase_Objects	<erase objects opc> <oin spec>
VEMMI_Open_Blocking_Object	<open block object opc> <oin>
VEMMI_Resource_Transfer	<transfer opc> <transfer entity>
VEMMI_Store_Objects_Response	<store objects resp opc> <store result>
VEMMI_Object_Retransmission	<object retrans opc> <oin>
VEMMI_Identify_Term_Cap_Resp	<term cap response opc> <cap response>+
VEMMI_Open_Application_Response	<open appl resp opc> <entity open appl resp>
VEMMI_User_Data	<user data opc> <entity user data>
VEMMI_Error	<error opc> <entity errorreport>
VEMMI_Resource_Transfer_Abort	<transfer opc> <transfer id>

<translation mode>:=	<2/0> { pas de conversion } <2/1> { codage 3 sur 4 } <2/2> { codage avec décalage en 7 temps }
<setup entity>:=	<version> <mode>
<version>:=	<INTEGER> { Version }
<mode>:=	<native> <page based>
<native>:=	<INTEGER : 0> { VEMMI telle que décrite dans la présente Recommandation }
<page based>:=	<INTEGER : 1> <basic page #> <input timeout>o
<basic page #>:=	<basic page # opc> <string>
<input timeout>:=	<input timeout opc> <INTEGER> { en secondes }
<cap request>:=	<data storage opc>o <local types opc>o <user language opc>o <system information opc>o
<cap response>:=	<version> <term cap list>
<term cap list>:=	<local storage info>o <data type info>o <user language>o <system information>o
<local storage info>:=	<local storage opc> <true> <false>
<data type info>:=	<data type opc> <text type> <still picture> <audio>o <graphic>o <moving picture>o <audio visual>o
<user language>:=	<user language opc>o <string> { Liste des codes linguistiques alphabétiques codés conformément à l'Annexe B/ISO 639 (exemple: langue néerlandaise: «NL») }
<system information>:=	<system information opc>o <string> <text type>:= <text type opc> <text type list>+ <end of list>
<still picture>:=	<still picture opc> <still picture list>+ <end of list>
<graphic>:=	<i>pas encore défini.</i>
<audio>:=	<audio opc> <audio type list>+ <end of list>
<moving picture>:=	<moving picture opc> <moving picture list>+ <end of list>
<audio visual>:=	<audio visual opc> <audio visual list>+ <end of list>
<text type list>:=	<INTEGER : 0> { Chaîne T.51 } <INTEGER : 1> { Texte de haute qualité VEMMI } <INTEGER : 2> { ISO 8859-1 } <INTEGER : 3> { ISO 10646-1 } <INTEGER : 4> { Inversion de code JIS } <INTEGER : 5> { ISO 8859-2 } <INTEGER : 6> { ISO 8859-3 } <INTEGER : 7> { ISO 8859-4 } <INTEGER : 8> { ISO 8859-5 } <INTEGER : 9> { ISO 8859-6 } <INTEGER : 10> { ISO 8859-7 } <INTEGER : 11> { ISO 8859-8 } <INTEGER : 12> { ISO 8859-9 } <INTEGER : 13> { ISO 8859-10 }

<error type>:= <INTEGER : 0> | { erreur générale }
 <INTEGER : 1> | { commande inconnue }
 <INTEGER : 2> | { commande erronée }
 <INTEGER : 3> | { erreur de syntaxe d'objet }
 <INTEGER : 4> | { commande inattendue }
 <INTEGER : 5> | { mémoire insuffisante }
 <INTEGER : 6> | { traitement des objets audio impossible }
 <INTEGER : 7> | { traitement des objets vidéo impossible }
 <INTEGER : 8> | { indice de couleur non valable }
 <INTEGER : 9> | { fichier pas trouvé }
 <INTEGER : 10> | { échec de la conversion à la phototrame }
 <INTEGER : 11> | { traitement de la définition col. direct. impossible }
 <INTEGER : 12> | { objet opératif refusé par l'utilisateur }
 <INTEGER : 13> | { mémoire locale insuffisante }
 <INTEGER : 14> | { un objet ouvert a été effacé }
 <INTEGER : 15> | { service pas traité }

<error oin>:= <error oin opc> <INTEGER>

<error cin>:= <error cin opc> <INTEGER>

<error com code>:= <error com code opc> <a/b> { a = 2..4, b = 0..15 }

<transfer entity>:= <data transfer> | <transfer abort>

<data transfer>:= <INTEGER : 0> <transfer id> <current block number> <current block data length>
 <block>

<transfer id>:= <INTEGER> { identificateur de transfert unique pour chaque fichier }

<current block number>:= <INTEGER> { de un jusqu'au nombre total de blocs }

<current block data length>:= <INTEGER>

<block>:= <file&block> | <current block>

<file&block>:= <filespec> <current block>

<filespec>:= <filename> <creation date> <file length> <total number of blocks>

<creation date>:= <string> { dans le format YYYYMMDDHHMMSS }

<file length>:= <INTEGER>

<total number of blocks>:= <INTEGER>

<transfer abort>:= <INTEGER : 1> <transfer id>

<current block>:= <current block opc> <current block data>

<current block data>:= <a/b>+ { a, b = 0..15 }

8.5 Objets, composants

<display object>:= <application bar> | <button bar> | <pop-up menu> |
 <dialogue box> | <operative object> | <sound object> |
 <video object> | <message box>

<application bar>:=	<appl bar opc> <application bar body> <menu item>+
<application bar body>:=	<xpos>o <ypos>o <height>o <width>o <closed>o <notaccessible>o <defactive>o <vertical>o
<menu item>:=	<bar menu choice> <pull down menu item>*
<bar menu choice>:=	<bar menu choice opc> <menu choice>
<pull down menu item>:=	<pull down menu choice> <cas menu choice>
<pull down menu choice>:=	<pull down menu choice opc> <menu choice> <separated>o
<cas menu choice>:=	<cas menu choice opc> <menu choice> <separated>o
<menu choice>:=	<cin> <height>o <width>o <notaccessible>o <text> <locactact>o <locactval>o
<button bar>:=	<button bar opc> <button bar body> <push button comp>+
<button bar body>:=	<xpos> <ypos> <width>o <height>o <closed>o <notaccessible>o <vertical>o <defactive>o <modal>o
<push button comp>:=	<push button comp opc> <cin> <height>o <width>o <closed>o <notaccessible>o <button> <locactact>o <locactval>o
<pop-up menu>:=	<pop-up menu opc> <pop-up menu body> <pop-up menu choice>+
<pop-up menu body>:=	<xpos> <ypos> <width>o <closed>o <notaccessible>o <title>o <title font>o <defactive>o <modal>o
<pop-up menu choice>:=	<pop-up menu choice opc> <menu choice> <separated>o <cas menu choice>*
<dialogue box>:=	<dialogue box opc> <dialog box body> <separator>* <frame>* <graphic output area>* <text area>* <text input field>* <box push button>* <check box>* <radio button>* <list box>* <combo box>* <slider>* <sensitive area>* <text component>* <sensitive text>*
<dialog box body>:=	<xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <no border>o <title>o <defactive>o <modal>o <maximizable>o <background>o <store ini values>o
<background>:=	<colour> <bitmapped>
<bitmapped>:=	<bin reference> <bitmap disptype>o <colour>o
<separator>:=	<separator opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <vertical>o <colour>o
<frame>:=	<frame opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <colour>o
<graphic output area>:=	<graphic output area opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <graphic type>
<graphic type>:=	<bitmap> <videotex>
<bitmap>:=	<bitmap disp type>o <bin reference>
<videotex>:=	<data syntax type> <direct videotex content> <vin reference>
<data syntax type>:=	<INTEGER : 0> { <i>Annexe B/T.101</i> } <INTEGER : 1> { <i>Annexe C/T.101</i> } <INTEGER : 2> { <i>Annexe D/T.101</i> }
<direct videotex content>:=	<videotex content opc> <nr of vtx bytes> <vtx bytes>+
<nr of vtx bytes>:=	<INTEGER> { <i>nombre d'octets vidéotex</i> }

<text area>:=	<text area opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <initial font>o <maxim text>o <no scrolling tools>o <no format>o <no border>o <autoscroll>o <text area content>o
<text area content>:=	<cin reference> { <i>cin d'un composant textuel</i> } <in-text>+
<text component>:=	<text component opc> <cin> <prev text>o <current text> <next text>o
<prev text>:=	<prev text opc> <cin reference> { <i>cin d'un composant textuel</i> }
<next text>:=	<next text opc> <cin reference> { <i>cin d'un composant textuel</i> }
<current text>:=	<current text opc> <tin reference> { <i>tin d'un objet ressource textuel</i> } <in-text>+
<in-text>:=	<text> <in-text string>
<in-text string>:=	<fin reference> { <i>fin d'un objet ressource de police</i> } <sensitive text reference>
<sensitive text reference>:=	<cin reference> <text> { <i>cin d'un composant textuel sensible</i> }
<sensitive text>:=	<sensitive text opc> <cin> <notaccessible>o <locactact>o <locactval>o
<text input field>:=	<text input field opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <maxim text>o <default text>o <label>o <label font>o <input type>o <echo type>o <echo char>o <max char>o <max line>o <multiline>o <input transform>o <locactact>o <locactval>o
<box push button>:=	<box push button opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <button> <locactact>o <locactval>o
<check box>:=	<check box opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <label>o <label font>o <marked>o <locactact>o <locactval>o
<radio button>:=	<radio button opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <label>o <label font>o <group>o <marked>o <locactact>o <locactval>o
<list box>:=	<list box opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <default item>o <sorted>o <multiple choice>o <locactact>o <locactval>o <list text unit>*
<combo box>:=	<combo box opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <default item>o <sorted>o <no drop>o <no consistency>o <no edit>o <locactact>o <locactval>o <list text unit>*
<list text unit>:=	<list index>o <text> <icon reference>o <item locact>o <item locval>o
<icon reference>:=	<bin reference>
<slider>:=	<slider opc> <cin> <xpos>o <ypos>o <height>o <width>o <closed>o <notaccessible>o <label>o <label font>o <negative>o <min value> <negative>o <max value> <increment>o <initial info>o <direct in>o <vertical>o <locactact>o <locactval>o
<initial info>:=	<negative>o <initial value>
<sensitive area>:=	<sensitive area opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <locator>o <locactact>o <locactval>o
<operative object>:=	<operative object opc> <closed>o <program name> <program filename> <program description> <program about> <program parameter>o <program type>
<sound object>:=	<sound object opc> <closed>o <filename> <sound format>
<sound format>:=	<INTEGER : 0> { <i>format wave</i> } <INTEGER : 1> { <i>format MIDI</i> }

<video object>:= <video object opc> <closed>o <filename> <moving picture list>o
<message box>:= <message box opc> <xpos>o <ypos>o <width>o <height>o <closed>o <no border>o <title>o <modal>o <message type>o <max time>o <lifetime>o <no sound>o <attributed text>+
<message type>:= <message type opc> <message>
<message>:= <INTEGER : 0> | { *message général* }
 <INTEGER : 1> | { *message d'information* }
 <INTEGER : 2> | { *message d'alerte* }
 <INTEGER : 3> | { *message d'action* }
<lifetime>:= <lifetime opc> <destroy event>
<destroy event>:= <INTEGER : 0> | { *détruit par toute action d'utilisateur* }
 <INTEGER : 1> | { *détruit par la validation du bouton défini implicitement* }
 <INTEGER : 2> | { *détruit par une commande du serveur* }
 <INTEGER : 3> | { *fermé par la validation du bouton défini implicitement* }
 <INTEGER : 4> | { *fermé par toute action d'utilisateur* }
 <INTEGER : 5> | { *aucune durée implicite définie* }
<max time>:= <max time opc> <INTEGER> { *durée en secondes* }

<resource object>:= <bitmap object> <text object> <videotex object>

<bitmap object>:= <bitmap object opc> <bitmap data> | <bitmap file>
<bitmap data>:= <INTEGER : 0> <bmwidth> <bmheight> <bmcompr>o <pixel definition>
<bmwidth>:= <INTEGER>
<bmheight>:= <INTEGER>
<bmcompr>:= <bmcompr opc> <INTEGER : 0> { *pas de compression* }
<pixel definition>:= <index def> | <rgb def>
<index def>:= <index def opc> <bits per pixel>o <colour entry>o <colour list>
<bits per pixel>:= <bits per unit>
<colour entry>:= <colour entry opc> <INTEGER>
<colour list>:= <list spec> { *bmwidth * bmheight (entiers)* }
<rgb def>:= <rgb def opc> <bits per component>o <component list>
<bits per component>:= <bits per unit>
<component list>:= <list spec> { *3 * bmwidth * bmheight (entiers)* }
<bits per unit>:= <bits per unit opc> <INTEGER>
<bitmap file>:= <INTEGER : 1> <filename> <pict file type>
<pict file type>:= <INTEGER : 0> | { *JPEG* }
 <INTEGER : 1> | { *GIF* }
 <INTEGER : 2> | { *BMP* }
:= <fn family>o <fn height>o <fn bold>o <fn underline>o <fn italic>o <fn colour>o
<fn family>:= <fn family opc> <family>

<family>:= <INTEGER : 0> | { *SWISS* }
 <INTEGER : 1> | { *ROMAN* }
 <INTEGER : 2> | { *FIXFONT* }
 <fn height>:= <fn height opc> <INTEGER>
 <fn bold>:= <fn bold opc>
 <fn underline>:= <fn underline opc>
 <fn italic>:= <fn italic opc>
 <fn colour>:= <fn colour opc> <INTEGER>
 <text object>:= <text object opc> <in-text>+ | <text file>
 <text file>:= <file name> <text file type>
 <text file type>:= <INTEGER : 0> | { *format «in-text»* }
 <INTEGER : 1> | { *texte, y compris CR et LF* }
 <videotex object>:= <videotex object opc> <vtx file>
 <videotex file>:= <filename> <data syntax type>

<metacode object>:= <metacode object opc> <command>+ <command end>
--

8.6 Actions locales

<locactact>:= <locactact opc> <local action>+
 <locactval>:= <locactval opc> <local action>+
 <local action>:= <report command> | <general command> | <specific command>
 <report command>:= <loc command opc> <report type>
 <report type>:= <INTEGER : 0> | { *communiquer les OIN, CIN* }
 <INTEGER : 1> | { *communiquer la valeur actuelle* }
 <INTEGER : 2> | { *communiquer toutes les valeurs* }
 <INTEGER : 3> | { *communication de toutes les valeurs modifiées* }
 <general command>:= <loc command opc> <general command type>
 <general command type>:= <INTEGER : 50> | { *blocage de l'utilisateur* }
 <INTEGER : 51> | { *sélection des valeurs initiales et des états de l'objet de rattachement* }
 <specific command>:= <loc command opc> <specific command type> <oin spec> | <cin spec>
 <specific command type>:= <INTEGER : 100> | { *ouverture des composants de l'objet de rattachement* }
 <INTEGER : 101> | { *fermeture des composants de l'objet de rattachement* }
 <INTEGER : 102> | { *ouverture d'objets* }
 <INTEGER : 103> | { *fermeture d'objets* }
 <INTEGER : 104> | { *passage des composants de l'objet de rattachement à l'état inaccessible* }
 <INTEGER : 105> | { *passage des composants de l'objet de rattachement à l'état accessible* }
 <INTEGER : 106> | { *effacement d'objets* }
 <INTEGER : 107> | { *ouverture d'un seul objet de blocage* }

9 Codage

9.1 Structure des commandes

L'entité de données d'image (PDE) d'une entité d'image (PE) achemine une ou plusieurs commandes VEMMI. La dernière commande VEMMI d'une PE peut être fractionnée et répartie sur plusieurs PE. Une telle répartition est indiquée par le premier octet de l'entité PDE, appelé «more data indicator» (*indicateur de données à suivre*). La syntaxe de la PDE est la suivante:

<PDE>:= <mdi> <command>+

<mdi>:= <2/14> | { la dernière commande n'est que partiellement contenue dans cette PE, se prolongeant dans la PE suivante; «MORE» }

<2/15> { la dernière commande est entièrement contenue dans la PE; «LAST» }

NOTE – La spécification de la grammaire ne rend pas compte de la répartition des commandes entre plusieurs PE.

Une ou plusieurs commandes VEMMI sont précédées de l'en-tête VEMMI. Cet en-tête sert à identifier les données VEMMI dans le train de données de l'application standard. On appelle entité de données une telle séquence de commandes VEMMI. La dernière commande d'une entité de données peut être répartie entre plusieurs entités de données. Cette répartition est indiquée par le MDI (more data indicator). Exemples:

Trois commandes contenues dans une entité de données:

En-tête VEMMI	MDI (dernière commande)	Entité de données		
		Commande 1	Commande 2	Commande 3

Trois commandes réparties sur deux entités de données, avec fractionnement d'une commande:

En-tête VEMMI	MDI (données à suivre)	Entité de données	
		Commande 1	Commande 2 (première partie)

En-tête VEMMI	MDI (dernière commande)	Entité de données	
		Commande 2 (dernière partie)	Commande 3

9.2 Structure des objets, des composants et des attributs

Structure des objets

Les objets présentent l'une des structures suivantes:

- a) <object code> <object body> <component>*
- b) <object code> <attribute>+
- c) <object code> <command>+

Structure des composants

<component code> <cin> <attribute>*

Structure des attributs

Les attributs présentent l'une des structures suivantes:

- a) <attribute code> { l'attribut est défini uniquement par son code };
- b) <attribute code, value> { l'attribut est défini par son code et par sa valeur };
- c) <attribute> { l'attribut est défini par un autre attribut };
- d) <attribute code, attribute> { l'attribut est défini par son code et un autre attribut }.

9.3 Codage des symboles terminaux

9.3.1 Codes opération

Les codes opération servent à identifier les parties des commandes VEMMI. On les utilise pour coder:

- les codes des commandes;
- les codes des objets;
- les codes des composants;
- le code MDI;
- les codes des attributs.

La notation des symboles terminaux est la suivante: <a/b> où a, b = 0..15 (notation hexadécimale).

9.3.2 Entiers

Ce format est utilisé pour coder les valeurs non négatives sans partie fractionnaire. Un entier consiste en une séquence d'un ou plusieurs octets.

Règles de codage:

- a) le bit 8 mis à 1 indique l'absence d'extension; il s'agit du dernier octet de la séquence;
- b) le bit 8 mis à 0 et le bit 7 mis à 1 indiquent une extension; il ne s'agit pas du dernier octet;
- c) la valeur codée est spécifiée:
 - 1) pour les entiers d'un seul octet, par les bits 7 à 1;
 - 2) pour les entiers de plusieurs octets, par les bits 6 à 1 de tous les octets sauf le dernier et par les bits 7 à 1 du dernier octet;
- d) la partie la plus significative de la valeur est codée dans le premier octet;
- e) la partie la moins significative de la valeur est codée dans le dernier octet.

Symboles utilisant des entiers:

<INTEGER>

<false>:= <INTEGER : 0>

<true>:= <INTEGER : 1>

Exemple: Représentation codée du nombre 37.

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
1	0	1	0	0	1	0	1
Pas d'extension	Bits indiquant la valeur des entiers						

Exemple: Représentation codée du nombre 1741.

Octet 1	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	0	1	0	0	1	1	0	1
	Extension		Bits de plus fort poids					

Octet 2	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	1	1	0	0	1	1	0	1
	Pas d'extension	Bits de plus faible poids						

9.3.3 Enumération

Ce type de données désigne une valeur d'une série de valeurs normalisées.

Ces valeurs sont codées sous forme d'entiers selon la notation suivante: <INTEGER : n> où n prend la forme d'une valeur décimale.

9.3.4 Chaînes

Ce type de données désigne une séquence de caractères présentant la structure suivante:

<string>:=	<latin string> <non latin string> <unicode string>
<latin string>:=	<character>+ <end of string>
<character>:=	<a/b> <c/d> <CR> <LF> <ESC> { a = 2..7, b = 0..15, c = 10, 15 caractères codés conformément à ISO 8859 ou à la chaîne T.51 }
<end of string>:=	<9/12> { ST (caractère de fin de chaîne) de ISO 6429 }
<CR>:=	<0/13>
<LF>:=	<0/10>
<non latin string>:=	<shift JIS string> <T.52 string>
<shift JIS string>:=	<a/b> <c/d> <e/f> <CR> <LF> { a = 2..7, 10..13; b = 0..15, c = 8,9,14,15; d = 0..15; e = 4..15; f = 0..15 }
<end of JIS string>:=	<1/11> <5/12> { ST (caractère de fin de chaîne) de ISO 6429 }
<T.52 string>:=	appelle un complément d'étude
<unicode string>:=	<uni char> <uni char> <end of unicode string> { chaîne conforme à ISO 10646-1 (BMP) }
<uni char>:=	<a/b> { a = 0..15, b = 0..15 }
<end of unicode>:=	<ST version> <FF version>
<ST version>:=	<0/0> <9/12>
<FF version>:=	<F/F> <F/F>

9.3.5 NDC

Ce type de données désigne les valeurs scalaires horizontales ou verticales de l'espace NDC. Une NDC est un nombre réel à virgule fixe compris entre 0,0 (inclusivement) et 1 (exclusivement).

Règles de codage:

- a) une valeur NDC est codée dans une séquence d'un ou de plusieurs octets;
- b) le bit 8 mis à 1 indique l'absence d'extension; il s'agit du dernier octet de la séquence;
- c) le bit 8 mis à 0 et le bit 7 mis à 1 indiquent une extension; il ne s'agit pas du dernier octet;
- d) la valeur codée est spécifiée:
 - 1) pour une valeur NDC d'un octet, par les bits 7 à 1;
 - 2) pour une valeur NDC de plusieurs octets, par les bits 6 à 1 de tous les octets sauf le dernier et par les bits 7 à 1 du dernier octet;
- e) l'emplacement de la virgule binaire n'est pas codé, mais est censé se trouver:
 - 1) pour une valeur NDC d'un octet, immédiatement à gauche du bit 7;
 - 2) pour une valeur NDC de plusieurs octets, immédiatement à gauche du bit 6 du premier octet;
- f) la partie la plus significative de la valeur est codée dans le premier octet;
- g) la partie la moins significative de la valeur est codée dans le dernier octet.

La notation du symbole terminal est la suivante: <NDC>.

Exemple: Représentation codée du nombre $0,3125_{10} = 0,0101_2$

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
1	0	1	0	1	0	0	0
Pas d'extension	Bits de la valeur NDC						

Exemple: Représentation codée de la valeur NDC $0,205078125_{10} = 0,001101001_2$

Octet 1	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	0	1	0	0	1	1	0	1
	Extension		Bits de plus fort poids					

Octet 2	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	1	0	0	1	0	0	0	0
	Pas d'extension	Bits de plus faible poids						

9.4 Attributs et symboles de niveau inférieur

<autoscroll>	:=	<6/9>
<attributed text>	:=	<text> <fin reference>
<autostore>	:=	<7/8>
<bin>	:=	<INTEGER>
<bin reference>	:=	<11/8> <bin>
<bitmap disp type>	:=	<11/11> <INTEGER : 0> { <i>allongé</i> } <INTEGER : 1> { <i>centré</i> } <INTEGER : 2> { <i>juxtaposé</i> }
<block index>	:=	<12/4> <INTEGER>
<button>	:=	<text> <bin reference>
<cin>	:=	<INTEGER>
<cin reference>	:=	<12/7> <INTEGER> { <i>indication d'un composant</i> }
<cin spec>	:=	<in spec>+
<closed>	:=	<6/8>
<colour>	:=	<11/10> <INTEGER> { <i>indice de couleur</i> }
<command end>	:=	<2/15>
<defactive>	:=	<11/7> <INTEGER> { <i>CIN désigné</i> }
<default item>	:=	<12/3> <INTEGER>
<default text>	:=	<text>
<direct in>	:=	<7/10>
<display character>	:=	<a/b> { <i>a = 2..15, b = 0..15</i> }
<echo char>	:=	<14/6> <display character>
<echo type>	:=	<12/0> <INTEGER : 0> { <i>écho local</i> } <INTEGER : 1> { <i>pas d'écho</i> } <INTEGER : 2> { <i>caractère défini par l'écho</i> }
<end of list>	:=	<3/15>
<filename>	:=	<14/8> <string>
<fin>	:=	<INTEGER>
<fin reference>	:=	<10/13> <INTEGER> { <i>indication d'un objet ressource de police</i> }
<group>	:=	<12/2> <INTEGER>
<height>	:=	<9/2> <NDC>
<in spec>	:=	<list spec> <range spec> <one in> { <i>définit une liste, une série ou un numéro d'identification unique</i> }
<list spec>	:=	<11/0> <INTEGER>+ <end of list>
<range spec>	:=	<13/0> <from> <to>
<from>	:=	<INTEGER>
<to>	:=	<INTEGER>
<one in>	:=	<13/5> <INTEGER>
<increment>	:=	<12/10> <INTEGER>
<initial font>	:=	<11/12> <INTEGER> { <i>le FIN désigné</i> }
<initial value>	:=	<12/11> <INTEGER>
<input type>	:=	<11/14> <INTEGER : 0> { <i>tout caractère</i> } <INTEGER : 1> { <i>alphabétique</i> } <INTEGER : 2> { <i>numérique</i> } <INTEGER : 3> { <i>alphanumérique</i> }
<input transform>	:=	<11/5> <INTEGER : 0> { <i>pas de transformation</i> } <INTEGER : 1> { <i>passage en minuscule</i> } <INTEGER : 2> { <i>passage en majuscule</i> }
<item locact>	:=	<7/13> <local action>+
<item locval>	:=	<7/14> <local action>+
<label font>	:=	<11/13> <INTEGER> { <i>le FIN désigné</i> }
<label>	:=	<14/5> <string>
<list index>	:=	<12/4> <INTEGER>
<locator>	:=	<6/0>
<marked>	:=	<7/1>
<max char>	:=	<12/1> <INTEGER>

<max line>	:=	<12/5>	<INTEGER>	
<max value>	:=	<12/9>	<INTEGER>	
<maxim text>	:=	<7/0>		
<maximizable>	:=	<6/14>		
<min value>	:=	<12/8>	<INTEGER>	
<modal>	:=	<6/12>		
<multiline>	:=	<6/15>		
<multiple choice>	:=	<7/2>		
<negative>	:=	<7/9>		
<no border>	:=	<6/13>		
<no consistency>	:=	<7/15>		
<no drop>	:=	<7/3>		
<no edit>	:=	<7/4>		
<no format>	:=	<7/6>		
<no scrolling tools>	:=	<7/5>		
<no sound>	:=	<7/11>		
<notaccessible>	:=	<6/11>		
<num blocks>	:=	<10/8>	<INTEGER>	
<oin>	:=	<INTEGER>		
<oin spec>	:=	<in spec>+	{ définit les identificateurs d'objet }	
<output height>	:=	<9/4>	<NDC>	
<program about>	:=	<string>		
<program description>	:=	<string>		
<program filename>	:=	<filename>		
<program name>	:=	<string>		
<program parameter>	:=	<program parameter opc>	<string>	
<program type>	:=	<13/1>	<INTEGER : 0>	{ programme autonome }
			<INTEGER : 1>	{ programme avec interface de filtrage }
<result>	:=	<true>	<false>	
<scrolling tools>	:=	<11/1>	<INTEGER : 0>	{ dépendant du terminal }
			<INTEGER : 1>	{ défilement deux colonnes par deux colonnes }
			<INTEGER : 2>	{ défilement deux rangées par deux rangées }
<slider value>	:=	<INTEGER>		
<store result>	:=	<true>	<false>	
<separated>	:=	<6/10>		
<sorted>	:=	<7/12>		
<template>	:=	<7/7>		
<text>	:=	<14/3>	<string>	
<timestamp>	:=	<12/14>	<INTEGER>	
<tin reference>	:=	<10/15>	<INTEGER>	{ indication d'un objet ressource de texte }
<title font>	:=	<11/9>	<INTEGER>	{ le FIN désigné }
<title>	:=	<14/4>	<string>	
<vertical>	:=	<6/7>		
<vin reference>	:=	<10/12>	<INTEGER>	
<width>	:=	<9/3>	<NDC>	
<xloc>	:=	<NDC>		
<xpos>	:=	<9/0>	<NDC>	
<yes/no>	:=	<true>	<false>	
<ypos>	:=	<9/1>	<NDC>	
<yloc>	:=	<NDC>		
<vtx byte>	:=	<a/a>	{ a = 0..15 }	

9.5 Codes opération

Les symboles de la spécification de la syntaxe des commandes présentés dans les Tableaux 39 et 40 sont dépourvus du suffixe «opc».

Les positions de code actuellement non utilisées sont réservées pour des extensions futures.

TABLEAU 39/T.107

Codes opération des commandes du serveur

	2	3
0	open	create object
1	close	delete outdated objects
2	resume	modify component (voir aussi 9.6)
3	suspend	obj location change
4	close all	load coltable
5	reset col	open application
6	user lock	store objects
7	user unlock	erase objects
8	open object	open blocking object
9	close object	resource transfer
10	destroy object	term cap request
11	access disable	
12	access enable	set options
13	translation mode	
14	(voir Note)	
15	(voir Note)	

NOTE – Ces positions de code sont réservées pour le MDI.

TABLEAU 40/T.107

Codes opération des commandes du terminal

	2	3
0	store objects resp	
1	object retrans	
2	open appl resp	
3	user data	
4	error	
5		
6	term cap resp	
7		
8		
9		resource transfer
10		
11		
12		
13	translation mode	
14		
15		

NOTE – Les Tableaux 41 à 43 contiennent les symboles qui ont le même code. Leurs noms sont séparés par une barre oblique («/»).

TABLEAU 41/T.107

Codes opération (Partie 1)

	Objets	Codes «End»	Composants	
	2	3	4	5
0	barre d'application		option barre de menu	ascenseur
1	barre de boutons		option menu déroulant	texte
2	menu incrusté		poussoir	texte sensible
3	boîte de dialogue		option menu incrusté	option menu empilé
4			séparateur	
5	objet opératif		cadre	
6	objet audio		zone d'affichage de graphiques	
7	objet vidéo		zone de texte	
8	boîte de messages		zone de saisie de texte	
9	objet de metacode		touche poussoir	
10	objet en phototrame		case d'option	
11	objet police		cercle d'option	
12	objet texte		boîte d'énumération	
13	objet vidéotex		boîte de combinaison	
14		fin de commande	zone sensible	
15		fin de liste		

TABLEAU 42/T.107

Codes opération (Partie 2)

	Attributs booléens		Coordonnées, dimensions
	6	7	8
0	fn bold/locator	maxim text	xpos
1	fn underline/local storage	marked	ypos
2	fn italic	multiple choice	height
3	index def	no drop	width
4	rgb def	no edit	output height
5	locactact	no scrolling tools	
6	locactval	no format	
7	vertical	template	
8	closed	autostore	
9	autoscroll	negative	
10	separated	direct in/more blocks	
11	notaccessible	no sound/store ini values	
12	modal	sorted	end of string
13	no border	item locact	
14	maximizable	item locval	
15	multiline	no consistency	

TABLEAU 43/T.107

Codes opération (Partie 3)

	Attributs d'entier				Attributs de chaîne
	10	11	12	13	14
0	fn family	list spec	echo type	range spec	applid/ user language
1	fn height/ data types	error oin/ scrolling tools	max char	program type	appl add data/ system information
2	fn colour/ text types	error cin	group	prev text	
3	bmcompr/ still picture	error com code	default item	current text	text
4	bits per unit	local commands	list index/ block index	next text	title
5	colour entry/ audio	input transform	max line	one in	label
6	moving picture				echo char
7	audiovisual	defactive	cin reference		vtx content
8	num blocks	bin reference	min value		filename
9	message type	title font	max value		program parameter
10	lifetime	colour	increment		current block
11	default vtx	bitmap disptype	initial value		
12	vin reference	initial font	max time		basic page #
13	fin reference	label font	input timeout		
14		input type	timestamp		
15	tin reference				

9.6 Syntaxe de la commande VEMMI_Modify_Component

La commande VEMMI_Modify_Component sert à modifier un ou plusieurs composants d'un objet ou à lui en ajouter un ou plusieurs. Cette commande a la même syntaxe que la commande VEMMI_Create_Object, sauf qu'elle ne contient que les attributs qui doivent être modifiés. On trouvera ci-dessous un état récapitulatif des attributs dont la définition des symboles supérieurs peut être modifiée:

autoscroll	echo type	list text unit	no sound
attributed text	group	locactval	notaccessible
bin reference	increment	locactact	previous text
cin reference	initial font	max char	text
closed	initial info	max value	tin reference
colour	input type	marked	title
current text	intext	message	
default item	label	min value	
default text	label font	negative	
echo char	lifetime	next text	

On modifie un attribut booléen en le transmettant une nouvelle fois. L'application assurée par le serveur peut sélectionner la valeur d'un attribut booléen ou annuler la sélection de cette valeur.

Pour sélectionner un attribut booléen, l'application assurée par le serveur doit transmettre l'attribut une fois (cette opération équivaut à la syntaxe de création d'objet). Pour annuler la sélection d'un attribut booléen, l'application assurée par le serveur doit transmettre cet attribut deux fois de suite.

Dans l'exemple de menu incrusté présenté ci-dessous, quatre options s'offrent à l'utilisateur, dont deux accessibles (CIN 35 et 36) et deux inaccessibles (CIN 37 et 38):

<u>S</u> tart New Game
<u>B</u> eginner Level
<u>I</u> ntermediate Level
<u>A</u> dvanced Level

A la réception de la commande VEMMI_Modify_Component structurée comme suit:

```
<oin : 2>  
<cin : 37> <NotAccessible> <NotAccessible>  
<cin : 39> <text : "&Double Cube"> <separated>
```

le menu est modifié, un composant inaccessible est rendu accessible et un nouveau composant est ajouté.

<u>S</u> tart New Game
<u>B</u> eginner Level
<u>I</u> ntermediate Level
<u>A</u> dvanced Level
<u>D</u> ouble Cube

Pour les attributs facultatifs, les valeurs par défaut sont appliquées si la commande VEMMI_Modify_Component ajoute un nouveau composant. La règle applicable ici est celle qu'utilise la commande VEMMI_Create_Object. Si la commande VEMMI_Modify_Component se rapporte à un composant qui existe déjà, seuls les attributs transmis sont appliqués. Les attributs manquants restent inchangés.

La commande VEMMI_Modify_Component ne doit pas être appliquée aux composants de la barre d'application.

Les conditions spécifiques applicables à la modification des items de liste sont décrites aux 7.6.1.11 et 7.6.1.12.

9.7 Valeurs par défaut

Les valeurs par défaut des attributs sont appliquées lorsqu'un attribut facultatif n'est pas contenu dans la définition de l'élément. Fait exception à cette règle, la commande VEMMI_Modify_Component qui est appliquée aux composants existants (voir 9.6). Les valeurs par défaut dépendent du type de données d'attribut et de l'objet/du composant dont elles font partie.

Pour la valeur par défaut d'un attribut booléen (dont le codage est effectué uniquement à l'aide du code opération), on prend la signification négative logique par rapport à sa définition. Une valeur par défaut d'un attribut booléen ne peut être codée et n'est jamais contenue dans la définition de l'élément. Voir les Tableaux 44 à 48.

TABLEAU 44/T.107

Valeurs par défaut des attributs booléens

Attribut	Valeur par défaut	Attribut	Valeur par défaut
fn bold	normal, non gras	maxim text	pas de texte plein écran
fn underline	non souligné	marked	non coché
fn italic	non italique	multiple choice	option unique
vertical	horizontal	no drop	déroulement
closed	ouvert	no edit	retraitable
notaccessible	accessible	no scrolling tools	avec outils de défilement
modal	non modal	no format	formatable
no border	avec bordure		
autoscroll	pas de défilement automatique	template	pas de mémorisation de «gabarit»
separated	non séparé	autostore	pas de «mémorisation automatique»
multiline	ligne unique	no sound	avec son
no consistency	avec cohérence	sorted	non classé
maximizable	agrandissement plein écran impossible	negative	positif
		direct in	pas de saisie directe

TABLEAU 45/T.107

Valeurs par défaut des attributs d'entier

Attribut	Valeur par défaut	Attribut	Valeur par défaut
fn family	SWISS	input type	0
fn height	10	echo type	0
fn colour	0 (noir)	max char	1
bits per units (pixel)	1	group	1
bits per units (component)	8	default item	1
		list index	1 pour le premier index, tous les autres étant numérotés en séquence
message type	0	max line	dépendant du terminal
lifetime	1	sound format	0
colour entry	0	initial value	valeur minimale
		increment	1
defactive	1	max time	10
title font	police par défaut	input timeout	infini
colour	0 (noir)		
bitmap disptype	allongé		
initial font	police par défaut		
label font	police par défaut		

TABLEAU 46/T.107

Valeurs par défaut des attributs NDC

Attribut	Valeur par défaut
xpos, ypos de la boîte de messages	dépendant du terminal
xpos	0,0
ypos	0,0

TABLEAU 47/T.107

Valeurs par défaut de largeur et de hauteur

Elément	Valeur par défaut de largeur	Valeur par défaut de hauteur
barre d'application	dépendant du terminal	dépendant du terminal
bouton: barre de boutons	0,125	0,03
menu incrusté	dépendant du terminal	
boîte de dialogue	1,0	0,75
séparateur	largeur de la boîte de dialogue	hauteur de la boîte de dialogue
cadre	largeur de la boîte de dialogue	hauteur de la boîte de dialogue
zone de texte	largeur de la boîte de dialogue	hauteur de la boîte de dialogue
zone d'affichage de graphiques	largeur de la boîte de dialogue	hauteur de la boîte de dialogue
zone de saisie de texte	largeur de la boîte de dialogue	0,03
poussoir	0,125	0,03
case d'option	0,125	0,03
cercle d'option	0,125	0,03
boîte d'énumération	largeur de la boîte de dialogue	hauteur de la boîte de dialogue
boîte de combinaison	largeur de la boîte de dialogue	hauteur de la boîte de dialogue
zone sensible	0,125	0,03
boîte de messages	dépendant du terminal	dépendant du terminal

TABLEAU 48/T.107

Valeurs par défaut des attributs de chaîne

Attribut	Valeur par défaut
echo char	2/13

10 Introduction du service VEMMI dans les Recommandations existantes concernant le vidéotex

10.1 Introduction du service VEMMI dans la Recommandation T.101 [4]

Les éléments de protocole VEMMI sont appliqués aux éléments de données de présentation vidéotex (VPDE).

Pour permettre la circulation des données entre le terminal vidéotex et l'application VEMMI, la voie logique doit être transparente pour les blocs de données VEMMI.

10.2 Introduction du service VEMMI dans la Recommandation T.105 [6]

Les éléments de protocole VEMMI sont appliqués à un ou plusieurs éléments de service de type SBV_VTX_Data.

Pour permettre la circulation des données entre le terminal vidéotex et l'application VEMMI, la voie logique doit être transparente pour les blocs de données VEMMI.

Annexe A

Chaîne T.51

(Cette annexe fait partie intégrante de la présente Recommandation)

A.1 Champ d'application

La présente annexe décrit les règles à appliquer pour le codage de noms, de chaînes, etc., selon la Recommandation T.51 [2]. La chaîne T.51 définie dans la présente annexe sert de modèle de référence en limitant la Recommandation T.51 [2] aux éléments des mécanismes d'extension de codes, des jeux de caractères et du répertoire qui sont nécessaires pour faciliter les mises en œuvre.

La présente définition n'est pas propre à un service de télématique particulier, mais peut être mentionnée dans des normes d'application de télématique.

A.2 Jeux de caractères graphiques

Le jeu primaire de caractères graphiques (voir la Figure A.1) est identique au jeu de caractères graphiques de la version internationale de référence (IRV) du jeu de caractères codés à 7 bits de la Recommandation T.50 [1].

				b ₇	0	0	0	0	1	1	1	1	
				b ₆	0	0	1	1	0	0	1	1	
				b ₅	0	1	0	1	0	1	0	1	
					0	1	2	3	4	5	6	7	
b ₄	b ₃	b ₂	b ₁										
0	0	0	0	0				0	@	P	.	p	
0	0	0	1	1				!	1	A	Q	a	q
0	0	1	0	2				"	2	B	R	b	r
0	0	1	1	3				#	3	C	S	c	s
0	1	0	0	4				\$	4	D	T	d	t
0	1	0	1	5				%	5	E	U	e	u
0	1	1	0	6				&	6	F	V	f	v
0	1	1	1	7				'	7	G	W	g	w
1	0	0	0	8				(8	H	X	h	x
1	0	0	1	9)	9	I	Y	i	y
1	0	1	0	10				*	:	J	Z	j	z
1	0	1	1	11				+	;	K	[k	{
1	1	0	0	12				,	<	L	\	l	
1	1	0	1	13				-	=	M]	m	}
1	1	1	0	14				.	>	N	^	n	~
1	1	1	1	15				/	?	O	_	o	

T0821490-95/d42

FIGURE A.1/T.107
 Jeu primaire de caractères graphiques pour la chaîne T.51

Le jeu supplémentaire de caractères graphiques est spécifié à la Figure A.2.

					b ₇	0	0	0	0	1	1	1	1
					b ₆	0	0	1	1	0	0	1	1
					b ₅	0	1	0	1	0	1	0	1
						0	1	2	3	4	5	6	7
b ₄	b ₃	b ₂	b ₁										
0	0	0	0	0			NBSP	◦		—	Ω	K	
0	0	0	1	1			ı	±	˘	¹	Æ	æ	
0	0	1	0	2			¢	²	´	®	Ð	đ	
0	0	1	1	3			£	³	^	©	ª	ð	
0	1	0	0	4				x	~	™	℥	℥	
0	1	0	1	5			¥	μ	—	♯		ı	
0	1	1	0	6				¶	˘	¬	IJ	ij	
0	1	1	1	7			§	.	·	ı	Ł	ł	
1	0	0	0	8			¤	÷	..		Ł	ł	
1	0	0	1	9			‘	’			Ø	ø	
1	0	1	0	10			“	”	°		Œ	œ	
1	0	1	1	11			«	»	˘		º	º	
1	1	0	0	12			←	¼		⅛	Ɔ	Ɔ	
1	1	0	1	13			↑	½	”	⅜	Ƨ	Ƨ	
1	1	1	0	14			→	¾	˘	⅝	Ƨ	Ƨ	
1	1	1	1	15			↓	ı	˘	⅞	’n	SHY	

T0821500-95/d43

Ces positions de codes ne doivent pas être utilisées

FIGURE A.2/T.107
 Jeu supplémentaire de caractères graphiques pour la chaîne T.51

Pour la chaîne T.51, les règles suivantes s'appliquent:

- 1) Le jeu primaire, désigné comme étant le jeu G0, est appelé dans les colonnes 2 à 7 de la grille de codes.
- 2) Le jeu supplémentaire est désigné comme étant un jeu G2. Dans l'environnement à 8 bits, il est appelé dans les colonnes 10 à 15.
- 3) Dans la chaîne T.51, les séquences de désignation et d'appel ne sont pas autorisées.
- 4) Tous les caractères de la colonne 4 du jeu supplémentaire sont des caractères sans espacement (signes diacritiques).
- 5) Les positions de codes non attribuées sont réservées et ne doivent pas être utilisées.
- 6) Les dispositions relatives à la compatibilité, décrites au 2.2.4/T.51 [2] (Notes 3, 4 et 6) ne sont pas appliquées.

A.3 Technique d'extension de code

Dans l'environnement à 8 bits, aucune séquence d'extension de code n'est autorisée. Dans l'environnement à 7 bits, la fonction d'inversion simple SS2 (code 1/9) est utilisée pour appeler un caractère du jeu G2. Toutes les autres fonctions d'inversion sont interdites.

A.4 Répertoire du jeu de caractères latins

Le répertoire de la chaîne T.51 est identique au superjeu du répertoire des caractères graphiques latins spécifié dans l'Annexe A/T.51 [2]. Toutes les combinaisons des signes diacritiques avec les lettres de l'alphabet de base spécifiées à la Figure A.2/T.51 [2] sont acceptées.

Les Notes et remarques générales du paragraphe A.4/T.51 [2] ne sont pas applicables.

A.5 Fonctions de commande

Les séquences d'appel et de désignation ne sont pas autorisées pour les fonctions de commande. D'après les colonnes 0 et 1 de la grille de codes en usage, seuls les caractères de commande CR, retour chariot (0/13) et LF, changement de ligne (0/10) peuvent être utilisés. Ils sont spécifiés dans la Recommandation T.50 [1].

Annexe B

Jeu secondaire obligatoire de ISO 8859 [13]

(Cette annexe ne fait pas partie intégrante de la présente Recommandation)

	2	3	4	5	6	7		10	11	12	13	14	15
0	SP	0	@	P	'	p							
1	!	1	A	Q	a	q							
2	"	2	B	R	b	r							
3	#	3	C	S	c	s							
4	\$	4	D	T	d	t							
5	%	5	E	U	e	u							
6	&	6	F	V	f	v							
7	'	7	G	W	g	w							
8	(8	H	X	h	x							
9)	9	I	Y	i	y							
10	*	:	J	Z	j	z							
11	+	;	K	[k	{							
12	,	<	L	\	l								
13	-	=	M]	m	}							
14	.	>	N	^	n	~							
15	/	?	O	_	o								

FIGURE B.1/T.107

Tableau de code du jeu secondaire obligatoire de ISO 8859 [13]

Annexe C

Noyau minimal de types de données

(Cette annexe fait partie intégrante de la présente Recommandation)

Un terminal VEMMI doit accepter le noyau minimal de types de données suivant.

TABLEAU C.1/T.107

Supported datatypes

Classe	Norme de codage	Observations
Données en mode texte		
	Texte de haute qualité VEMMI	
	ISO 8859 [13] (caractères préformés avec accent) de la position de code 2/0 à la position de code 7/15, comme indiqué dans l'Annexe B	Pour terminal à caractères latins
	Inversion de code JIS (pour caractères japonais)	Pour terminaux japonais
Images fixes		
	ISO 10918 [16] (JPEG)	
	Phototrame tributaire du type d'unité à VEMMI	
	Graphics Interchange Format (sm) Version 89a. CompuServe Incorporated Columbus, Ohio, USA (seules les fonctions de la Version 87 doivent être acceptées)	
	Phototrame Microsoft Windows non tributaire du type d'unité (DIB) avec RLE4, RLE8 et 1, 4, 8 ou 24 bits par pixel	
Données audio		
	Format WAVE	(voir Note)
	Format MIDI	(voir Note)

NOTE – Le qualificatif «obligatoire» signifie dans le cas présent qu'un terminal doit prendre en charge tous les services VEMMI se rapportant aux fonctions, et non pas que ce terminal doit être doté des équipements nécessaires pour lire ces données.