

Groupe de travail réseau
Request for Comments : 965
 Traduction Claude Brière de L'Isle

Lorenzo Aguilar
 SRI International
 décembre 1985

Format pour un protocole de communication graphique

Statut de ce mémoire

Le présent document décrit les exigences pour un format graphique sur lequel fonder un protocole de communication graphique en ligne. La proposition est un format de communication graphique interactif qui utilise le métafichier de session GKSM. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Le présent document décrit les exigences pour un format graphique sur lequel fonder un protocole de communication graphique en-ligne. Il est expliqué que la communication graphique en ligne est similaire à la capture de session graphique, et qu'on propose donc un format de communication graphique interactive utilisant le métafichier de session GKSM.

On expose les éléments estimés complémentaires du métafichier GKSM comme format pour les échanges interactifs en ligne. Un domaine d'application clé d'un tel format est une conférence multimédia en ligne ; on présente donc une architecture de logiciel de conférence pour traiter le format proposé. On rend cette spécification de format disponible pour ceux qui prévoient des systèmes de conférence multimédia comme une contribution au développement d'un protocole de communication graphique qui va permettre l'interopération de ces systèmes.

On espère que cette contribution encouragera la discussion des échanges de données multimédia et la proposition de solutions. À SRI, nous restons ouverts à l'exploration de solutions de remplacement et nous continuerons nos travaux de recherches et de développement dans ce domaine.

Remerciements

Les auteurs tiennent à remercier Andy Poggio de SRI de ses nombreuses suggestions précieuses qui ont structuré et amélioré le niveau U. Son expertise dans les systèmes de communication multimédia et ses encouragements ont été un apport très positif à la création de cet IGCF. Dave Worthington de SRI a aussi participé aux discussions du projet autour de cet IGCF. Des remerciements sont aussi dus à Tom Powers, président du groupe de travail ANSI X3H33, qui a ouvert ce forum à la présentation d'une version précédente du présent document, fournissant par là une opportunité pour les inestimables retours des membres du X3H33. Jon Postel de ISI a recommandé un certain nombre de changements qui ont rendu le document plus cohérent et accessible.

La plus grande partie du travail figurant dans le présent document a été financée par la U.S. Navy, Naval Electronic Systems Command, Washington D.C., sous le contrat n° N00039-83-K-0623.

Table des Matières

1. Introduction.....	2
1.1 Utilisation d'un protocole de communication graphique.....	2
1.2 Relations avec d'autres travaux.....	3
2. Exigences de fonctionnement et d'utilisation.....	4
2.1 Interopération de systèmes hétérogènes.....	4
2.2 Capture d'image.....	4
2.3 Prompte transmission.....	4
2.4 Faible volume de trafic.....	5
2.5 Préservation des unités sémantiques d'application.....	5
2.6 Mise en lots de transmission.....	5
2.7 Traduction simple entre IGCF et interface d'utilisateur.....	6
3. Éléments d'un IGCF.....	6
4. Proposition d'IGCF.....	7
4.1 Fondé sur l'utilisation de GKSM.....	7
4.2 Coordonnées des informations de position.....	8
4.3 Couches du PIGCF.....	8
4.4 Éléments PIGCF dans le niveau U.....	8
5. Architecture pour le traitement de PIGCF.....	10
6. Conclusions.....	12
Appendice A Extraits de "Projet de proposition de système de noyau graphique" [14].....	13
Appendice B Exemple d'utilisation de PIGCF en conférence.....	21
Références.....	23

1. Introduction

1.1 Utilisation d'un protocole de communication graphique

Dans le champ des communications informatiques, un protocole est une procédure exécutée par deux processus qui coopèrent afin d'atteindre à un échange significatif d'informations. Un protocole de communication graphique est nécessaire pour échanger des informations graphiques de vecteurs interactifs, éventuellement en conjonction avec d'autres supports d'information comme la voix, le texte, et la vidéo. Au sein de cet environnement de communication multimédia, les vecteurs graphiques informatiques jouent un rôle clé parce qu'ils tirent pleinement parti des capacités de traitement des ordinateurs communicants et des utilisateurs humains, et sont donc beaucoup plus compacts que les images numériques qui ne sont pas générées à partir de structures de données contenant des informations de position. La communication graphique vectorielle échange une utilisation intensive de la mémorisation et du traitement, aux extrémités de la communication, contre un faible volume de données échangées, parce que les stations de travail qui ont un matériel graphique échangent les commandes graphiques en conjonction avec de larges structures de données chez l'émetteur et les receveurs. De cette manière, la transmission d'une seule commande peut produire des changements extensifs entre les données affichées chez les extrémités d'envoi et de réception.

Il est utile de situer le protocole susmentionné dans les niveaux fonctionnels du modèle de référence d'interconnexion des systèmes ouverts de l'ISO [1]. Au sein d'un tel modèle, une fonctionnalité de protocole graphique appartient principalement au niveau application, bien qu'une partie relève du niveau présentation. On peut distinguer les composants suivants dans un protocole de communication:

- a) un format de données
- b) des règles pour interpréter les données transmises
- c) des tableaux d'informations d'état
- d) des règles d'échange de messages

Un format pour un protocole graphique devrait fournir la disposition des données transmises, et indiquer comment les données formatées sont associées aux règles d'interprétation. Le choix d'un format influence les tableaux d'état à tenir pour le traitement correct du flux de données transmises. Le format graphique a une influence mineure sur les règles d'échange, qui devraient assurer l'utilisation efficace de la capacité de transmission pour transporter les données sous un tel format. À côté du format graphique, il y a d'autres aspects d'un protocole graphique qui déterminent les tableaux d'état et les règles d'échange. Le présent article se concentre sur le format des données, et il ne traite pas de l'échange de messages. Néanmoins, on discute de l'architecture logicielle simple pour générer et interpréter les flux de données écrites dans notre proposition de format. De plus, on donne un exemple d'une application du format proposé (dans l'Appendice B) et il illustre le type des échanges de messages qui sont nécessaires pour établir une session de communication et d'échange des informations graphiques.

Ceux qui sont dans la communication informatique sont bien au courant de l'importance de protocoles largement acceptés pour réaliser une communication significative. Ceux qui ont besoin de mettre en œuvre aujourd'hui des communications graphiques interactives sont confrontés au manque de normes pour la communication graphique informatique dans les programmes d'application. Néanmoins, on peut utiliser certains des travaux déjà réalisés par les organismes de normalisation graphique informatique. En fait, l'ISO et l'ANSI ont déjà ajouté à la norme du système de noyau graphique (GKS, *Graphical Kernel System*) la spécification du méta-fichier de session GKSM qui possède beaucoup des caractéristiques nécessaires pour un protocole graphique en ligne.

Il est pertinent de mentionner un exemple de communication graphique qui illustre la nature en temps réel de l'interaction ainsi que l'utilisation des graphiques en conjonction avec d'autres supports d'informations. Avec la conférence audio-graphique, un groupe d'individus en deux localisations ou plus peut tenir une réunion électronique. Ils peuvent converser sur des canaux vocaux et partager concurremment un espace graphique sur lequel ils peuvent afficher, pointer sur, et manipuler, des images de vecteurs graphiques ; [2], [3], [4], [5], [6], [7].

Les canaux vocaux de conférence peuvent être fournis par diverses technologies de transmission. L'espace graphique partagé peut être mis en œuvre sur des stations de travail qui affichent les images et permettent l'interaction et la communication graphique avec les autres localisations. La communication des opérations sur des images implique des modifications aux structures de données sous-jacentes, mais on est concerné par la mise à jour de base de données graphique dans la seule mesure où une telle mise à jour prend en charge la communication.

Afin d'exécuter une session graphique enregistrée, on aura besoin d'indications sur le taux de représentation des éléments graphiques et de recreation des opérations graphiques. On n'inclut pas les moyens d'indiquer le rythme d'insertion d'une session dans un format parce que notre principal objet est de l'utiliser dans des environnements de communication à supports mixtes. Dans de tels environnements, le rythme d'exécution doit être compatible entre les supports d'information

afin de les coordonner. Donc, on laisse les mécanismes de rythme aux modules de contrôle de conférence. On laisse aussi aux processus de contrôle de conférence la manière dont une station de conférence émule une capacité graphique dont elle manque. Un exemple en est la représentation des couleurs dans un affichage monochrome.

1.2 Relations avec d'autres travaux

Il y a un certain nombre de normes actuelles et proposées pour l'échange d'informations graphiques. Dans ce qui suit, on explique les raisons pour lesquelles, jusqu'à présent, aucune d'entre elles ne peut être utilisée comme base d'un protocole en ligne. Comme certains de ces standard évoluent, certains peuvent cependant devenir convenables. De plus, l'expérience acquise avec les premiers systèmes de communication graphique en ligne fournira des indications sur les extensions appropriées de ces standard pour la prise en charge de systèmes plus évolués. De telles indications pourraient aussi être utilisées pour modifier le format proposé dans le présent mémoire, qu'on considère comme une approche initiale du problème. À l'avenir, le format proposé dans ce mémoire pourrait être remplacé par une des normes étendues susmentionnées.

La syntaxe nord américaine de protocole de niveau présentation (NAPLPS, *North American Presentation Level Protocol Syntax*) spécifie une syntaxe de données et une sémantique d'application pour les services unidirectionnels de diffusion d'informations télétexte et les services bidirectionnels d'accès et de transaction de base de données vidéotex. Le modèle de fonctionnement bidirectionnel du vidéotex se fonde sur le concept d'un consommateur et d'un fournisseur d'informations ou opérateur du service. À cause de cette asymétrie, on suppose que presque toutes les informations graphiques vont s'écouler du fournisseur vers le consommateur. Dans la direction inverse, le consommateur est supposé manipuler et transmettre essentiellement des informations alphanumériques. Bien que cette norme comporte des primitives de dessins géométriques, un utilisateur ne peut pas modifier directement des formes dessinées avec les primitives.

À présent, NAPLPS n'inclut pas de concept d'interaction comme des transformations d'image ou de capacité de détection, qui sont fondamentales pour atteindre un espace de travail graphique partagé, pas plus qu'il ne permet des appareils d'entrée graphique essentiels comme les souris, les joysticks, les stylets, les boules tournantes, ou les stylos électroniques, qui sont nécessaires pour une édition simple et efficace de l'espace de travail partagé.

On veut avoir une communication graphique d'usager à usager qui représente le niveau de sophistication et de facilité d'interaction fourni par les paquetages graphiques interactifs d'aujourd'hui. Les graphiques de vecteur informatique peuvent fournir les deux parce que leur paradigme inclut un programme d'application qui garde trace d'un très grand nombre de changements d'états possibles de l'image affichée. De plus, l'application pilote un paquetage graphique puissant, comme GKS ou ACM Core. Dans le paradigme du vidéotex, l'application fournisseur ne permet que des changements limités à l'image affichée, principalement des demandes de restitution de base de données. Aussi, le paradigme n'inclut pas un paquetage graphique séparé. Les deux fonctionnalités graphiques et un format sont renfermés dans une spécification de codage, comme NAPLPS.

Dans le présent article, on s'intéresse principalement aux applications d'affaires et industrielles où il y a un flux bidirectionnel ou multidirectionnel d'informations de vecteurs graphiques entre les utilisateurs. Les usagers auront des stations de travail avec des capacités substantielles de traitement et de mémorisation, et des moniteurs à haute résolution ; de plus, la communication sera sur une architecture répartie ne dépendant pas d'un hôte serveur central, comme l'hôte fournisseur d'application du vidéotex.

Actuellement, l'équipement de vidéotex chez le consommateur, consiste en des décodeurs fondés sur un microprocesseur à bas prix ou un écran d'ordinateur personnel pilotant, dans la plupart des cas, un poste de télévision standard à faible résolution et un affichage d'ordinateur personnel. Il y a déjà des technologies abordables pour produire des décodeurs sophistiqués et des appareils graphiques à haute résolution. Le vidéotex standard a besoin de révisions extensives pour tirer parti de ces avancées ; en particulier, il devrait considérer les appareils récepteurs comme capables d'héberger un processus programmable d'applications d'utilisateur. Quand cela arrivera, les protocoles vidéotex seront applicables aux domaines qui nous intéressent [8].

Le métafichier d'ordinateur graphique [9] va devenir un standard nord-américain et international pour les échanges d'images graphiques dans un avenir proche. Cependant, le CGM, aussi appelé VDM, est un métafichier de capture d'image qui n'enregistre que le résultat final d'une session graphique. Il n'est pas destiné à enregistrer le processus de création d'image, qui est fondamental pour les applications interactives que nous visons. De plus, le CGM vise présentement à une prise en charge minimale de la fonctionnalité GKS. Il faudra un certain temps avant que le CGM ait quelques uns des éléments nécessaires pour une interaction en ligne. Si, après ces ajouts, le CGM est agrémenté de la capture de session, il deviendrait un candidat logique pour un format de protocole.

Un autre futur standard est l'interface d'ordinateur graphique (CGI, *Computer Graphics Interface*) aussi appelé VDI [10]. Le CGI est un standard fonctionnel et une spécification syntaxique de l'échange de commandes et de données entre un

logiciel graphique indépendant de l'appareil et un ou plusieurs pilotes d'appareils graphiques dépendants de l'appareil. Une utilisation majeure de CGI est pour la communication entre un hôte d'application et un appareil graphique, mais l'asymétrie entre ses extrémités communicantes prévues empêche l'utilisation de CGI pour nos besoins.

Comme on l'a déclaré précédemment, on veut tirer parti de l'intelligence et des capacités de mémorisation des extrémités communicantes afin de réaliser de puissants effets de transports d'informations en utilisant des canaux à bande passante étroite. Cela exige que le format qu'on recherche ait des éléments pour communiquer entre deux applications. À l'opposé, les flux de CGI sont traités par des pilotes dépendants de l'appareil, plutôt que par les applications. La spécification CGI inclut des éléments de données d'application, mais seulement pour les mémoriser dans un métafichier. Ces éléments de données d'application ne sont pas interprétés par les pilotes, mais par les applications qui lisent le métafichier, un certain temps après la création du métafichier.

De plus, le CGI a des éléments pour obtenir des entrées graphiques, ainsi que des éléments pour s'enquérir des capacités de l'appareil graphique, de ses caractéristiques, et des états. Plus loin, dans la Section 3, on explique pourquoi ces deux classes d'éléments sont inutiles pour le protocole de communication dont nous avons besoin. Avec l'évolution du CGI, il va subir des changements significatifs, et, à l'avenir, il peut devenir un noyau très convenable pour le protocole graphique que nous recherchons. En fait, le CGI sera le protocole de communication entre les hôtes d'application graphique et les terminaux graphiques. À SRI, nous suivons son évolution, et sommes intéressés par la définition d'un format fondé sur le CGI.

Finalement, la spécification d'échange graphique initial [11] ne vise pas notre principal domaine d'intérêt. L'IGES définit des formats de fichier et de langage standard pour mémoriser et transmettre des données de définition de produit qui peuvent être utilisées, en partie, pour générer des dessins d'ingénierie et d'autres représentations graphiques de produits d'ingénierie. À côté de l'orientation DAD de IGES, la fonction de résultat graphique peut être secondaire par rapport à d'autres objectifs tels que de transmettre des instructions de machine à commande numérique.

2. Exigences de fonctionnement et d'utilisation

Le principal objectif de cet article est de poser les fondements pour le développement d'un format de vecteur graphique à utiliser pour un protocole en ligne de communication graphique. On appelle un tel format un "format de communication graphique interactive" (IGCF, *Interactive Graphical Communication Format*). On décrit dans la présente section certaines des exigences de fonctionnement et des caractéristiques d'utilisation d'un IGCF.

2.1 Interopération de systèmes hétérogènes

Une première exigence fonctionnelle est qu'un IGCF doit permettre la communication entre des systèmes graphiques hétérogènes différant tous deux par le matériel utilisé et le logiciel de leurs interfaces d'application graphique. Ceci est fondamental pour arriver à communiquer entre des programmes d'application graphique similaires fonctionnant sur des matériels différents et utilisant des paquetages d'interface graphique différents. Certains exemples de tels programmes d'application sont des éditeurs graphiques, des systèmes de CAO, et des programmes de restitution de base de données graphiques qui communiquent avec d'autres éditeurs, respectivement, des programmes de CAO, et des bases de données graphiques.

2.2 Capture d'image

Une caractéristique exigée d'un IGCF est qu'il doit être utilisable pour l'échange d'images graphiques statiques, c'est-à-dire, pour la capture d'image ; donc, il ne doit pas se restreindre seulement à l'enregistrement de l'image finale. Il y aura des échanges d'images au titre de la communication interactive, et on prévoit un besoin d'enregistrer l'état d'une image à différents points durant l'engagement graphique en ligne. On prévoit la création de bibliothèques d'IGCF graphiques qui contiendront des définitions d'objets et des images à inclure dans de nouvelles images. Comme des métafichiers ont été utilisés depuis longtemps pour capturer les images, il y a des fortes motivations pour appuyer un IGCF sur un standard de métafichier afin d'assurer la compatibilité avec un grand nombre de sources et de consommateurs de métafichiers.

2.3 Prompte transmission

Dans certaines formes de communication graphique interactive, comme la conférence audiographique, il est critique de transporter entre les utilisateurs la nature en temps réel de l'interaction. Cela impose que les créations et manipulations d'objets soient transmises lorsque elles se produisent plutôt que comme résultat final, car une partie substantielle de l'information peut être transmise concurremment avec la construction ou le fonctionnement d'un objet, éventuellement à travers des supports associés comme la voix. Comme les deux processus de construction et de manipulation doivent être transmis, il y a une limite au nombre d'états intermédiaires qui, du point de vue économique, peuvent être transmis.

Une troisième exigence est donc que les éléments de l'IGCF fournissent une fine "granularité" pour porter la dynamique des constructions et manipulations. On pense qu'il est suffisant que l'IGCF ait les éléments de construction de base comme les polygones, les marqueurs, les lignes multiples, et les chaînes de texte et qu'il ne les transmettent que lorsqu'elles sont terminées ; c'est-à-dire, il n'est pas nécessaire de transmettre des constructions partielles de tels éléments.

Le problème des manipulations s'étend au delà d'un IGCF. Bien qu'on sache qu'un IGCF devrait inclure des transformations de segment, le soulignement de segment et l'activation/désactivation de visibilité de segment, l'émetteur doit décider à quelle fréquence échantillonner une transformation en cours et transmettre son état actuel. Le choix d'une fréquence d'échantillonnage va dépendre de la bande passante de transmission disponible.

2.4 Faible volume de trafic

Dans beaucoup des applications qu'on envisage, les coordonnées graphiques seront transmises sur des canaux à bande passante étroite, et il est donc essentiel de minimiser le trafic. En conséquence, plusieurs exigences sont imposées à un IGCF pour tirer parti des caractéristiques des rapports et de l'architecture de la communication graphique afin de minimiser le trafic.

Un IGCF peut aider à réduire le trafic en incluant les objets géométriques de base à partir desquels de nombreux autres objets sont construits. De plus, un IGCF devrait permettre d'utiliser les objets pour la création d'objets plus complexes, car lorsque la réutilisation est très courante, il en résulte une réduction du trafic et des coûts de mémorisation.

2.5 Préservation des unités sémantiques d'application

Une exigence en rapport avec celle-ci est qu'un IGCF doit comporter des éléments pour représenter des objets graphiques correspondant à des entités réelles du monde des applications visées. Par exemple, dans une application de la Marine, les entités intéressantes sont des transporteurs, des sous-marins, des avions, et ainsi de suite. On veut communiquer de telles unités sémantiques à travers les systèmes et les traiter comme des objets unitaires parce que, dans de nombreuses applications, la communication se fonde sur la création et le fonctionnement de telles unités. Si un IGCF a des éléments pour représenter de telles unités sémantiques, le trafic de communication diminue parce que les définitions d'entités peuvent être transmises seulement une fois et ensuite réutilisées, et parce que les entités sont manipulées comme des unités plutôt que de manipuler séparément leurs composants.

Il apparaît qu'il y a un petit ensemble d'opérations primaires qui s'appliquent à un objet graphique, et qu'un IGCF doit avoir des éléments qui représentent de telles opérations. À l'opposé des terminaux graphique aveugles qui reçoivent des informations de rafraîchissement d'écran d'un hôte, on prévoit une communication graphique ayant lieu entre des stations de travail intelligentes qui peuvent échanger des opérations codées, les interpréter, et les appliquer aux objets mémorisés en local.

2.6 Mise en lots de transmission

On a indiqué précédemment qu'il est souhaitable d'apporter aux utilisateurs humains le rythme en temps réel des échanges graphiques interactifs. Il est cependant possible de le faire sans avoir à transmettre immédiatement tous les éléments d'IGCF. En fait, les éléments d'IGCF devraient être divisés selon qu'ils causent ou non un changement de l'affichage d'une image, bien que les deux classes puissent causer des changements aux structures de données graphiques mémorisées.

Il est seulement nécessaire de transmettre immédiatement les éléments qui causent un changement visible sur une image affichée parce que ce sont ceux dont la réception et l'interprétation livrent des informations à l'utilisateur humain. La seconde classe d'éléments peut être mise en lots et en file d'attente de transmission jusqu'à ce qu'un élément de la première classe soit soumis. On appelle Groupe-1 la mise à jour de la première classe, et Groupe-2 la mise à jour de la seconde.

La division susmentionnée est assez importante pour les communications par paquets parce que chaque paquet contient une grosse quantité de trafic de contrôle redondant. Il est donc obligatoire de faire des lots, dans un paquet, d'autant de données de client que possible afin de réduire le trafic total. Les unités qui constituent les lots peuvent varier en taille selon le trafic du réseau et le délai de réponse des hôtes en conférence. Durant les périodes d'encombrement, les unités peuvent devoir être augmentées, diminuant ainsi le nombre de messages, et ensuite être réduites lorsque l'encombrement se résorbe, augmentant donc le nombre de messages.

2.7 Traduction simple entre IGCF et interface d'utilisateur

Selon la première exigence, un IGCF doit permettre l'interopération des applications graphiques hétérogènes en cause. Une telle interopération a pour objectif la communication entre les utilisateurs humains ou entre un homme et une base de

données. De façon correspondante, l'interopération implique une transposition entre les commandes d'interface d'utilisateur et les éléments d'IGCF. Il n'est pas conseillé d'utiliser les commandes elles-mêmes comme éléments d'IGCF ; car l'échange dépendrait alors des systèmes communicants, et chaque paire de système communicant exigerait un protocole ad hoc.

Une caractéristique d'utilisation supplémentaire est qu'il doit y avoir une transposition simple entre les éléments d'IGCF et les actions représentées par les commandes d'interface d'utilisateur employées pour les communications graphiques. Cette simplicité est une obligation parce que tout système graphique communiquant doit avoir un traducteur qui dans l'idéal devrait être très simple. Il semble que l'inclusion de délimiteurs de séquence de commandes dans l'IGCF aide à la simplicité car les délimiteurs permettent de garder une plus petite quantité d'informations d'état pour traiter un flux d'IGCF.

On a vérifié la transposition d'un ensemble de commandes pour une conférence audiographique dans l'IGCF proposé dans le présent article. La transposition des commandes d'interface d'utilisateur en IGCF peut être faite de façon directe et efficace ; d'un autre côté, la transposition inverse, d'IGCF en commandes d'interface d'utilisateur, est une tâche plus difficile. On prévoit que, afin d'améliorer les performances, on aura à transposer les éléments d'IGCF en appels à des sous-programmes de niveau inférieur mettant en œuvre les actions de l'interface d'utilisateur. Bien que de telles transpositions ne soient pas conceptuellement plus complexes que de traduire l'IGCF en les commandes elles-mêmes, cela va exiger considérablement plus de programmation.

3. Éléments d'un IGCF

Classes d'élément IGCF

Dans cette section, on fait la liste des classes d'éléments qu'on pense qu'un IGCF devrait avoir afin d'échanger des vecteurs graphiques sous les exigences de la section précédente. Les classes correspondent aux classes de fonctions courantes sur les interfaces graphiques d'ordinateurs, et contiennent chacune les éléments qui correspondent aux primitives et attributs d'interface. On ne fait pas la liste des éléments pour chaque classe parce que ils sont représentés par les éléments de l'IGCF proposé.

Dans la liste suivante, deux catégories de fonctions manquent : les fonctions utilisées pour interroger l'état d'un système graphique, et les fonctions d'entrée. En fait, un IGCF a seulement besoin d'avoir des éléments qui représentent les actions qui causent un changement dans l'état des systèmes graphiques communicants, et les fonctions d'interrogation ne changent évidemment pas leur état. Même si une fonction d'entrée exécutée à l'extrémité d'émission cause un changement local, il n'est pas nécessaire de transmettre la commande d'entrée elle-même. Les receveurs ont seulement besoin d'obtenir l'entrée des données, en représentation IGCF, et ils peuvent traiter les données de toutes les façons, peut-être en simulant les actions d'entrée locales.

Contrôle

Ce sont les éléments pour la station de travail : initialisation, contrôle et transformation, et les éléments pour la transformation de normalisation. (La normalisation et les transformations de station de travail peuvent être utilisées pour mettre en œuvre le grossissement.)

Attributs de primitive

Ce sont les éléments pour les attributs de primitive, de segment, et de station de travail.

Primitives de sortie

Ce sont les éléments pour les primitives de sortie.

Segmentation

Ce sont les éléments pour la segmentation de base et la mémorisation de segment indépendante de la station de travail. Les manipulations d'objet peuvent être mises en œuvre avec des transformations de segment. L'insertion d'objet peut être mise en œuvre en utilisant le rappel de segment et la visibilité de segment. La suppression d'objet peut être mise en œuvre en utilisant la suppression de segment et la visibilité de segment. La sélection d'objet peut utiliser la mise en lumière de segment comme retour pour l'utilisateur.

Dynamiques

Une partie considérable des informations graphiques échangées dans un IGCF seront sous forme de mouvements de pointeur sur une figure en arrière plan. Le suivi de pointeur est utilisé pour transmettre des points échantillonnés à partir d'une trace de pointeur graphique afin de reproduire, chez les receveurs, le mouvement du pointeur au site de l'expéditeur. Cela peut être fait soit juste en déplaçant le curseur, soit en traçant son mouvement avec une ligne. Les échos de bande réfléchissante sont utilisés pour signaler des zones, des routes, et des portées de façon très dynamique. Ils sont indiqués par

un point de référence d'écho et un point de rétroaction.

Définitions d'objet hiérarchique

Les exigences pour la préservation de la sémantique des applications imposaient qu'un IGCF inclue le moyen de représenter des objets qui tiennent la place d'entités d'application, et de manipuler de telles entités comme des unités graphiques. De plus, l'exigence de faible volume de trafic impose l'utilisation d'objets déjà existants pour la création de nouveaux.

Une façon de satisfaire les exigences susmentionnées est d'inclure dans un IGCF le moyen de représenter les hiérarchies d'objets. Dans une telle hiérarchie, un objet est un ensemble de primitives de sortie associées à un ensemble de valeurs d'attribut ou à un ensemble d'objets de niveau inférieur, associé chacun à une composition de transformations [12].

Les segments graphiques peuvent être utilisés pour mettre en œuvre des objets dans le plus bas niveau d'une hiérarchie. La définition d'un objet de niveau supérieur peut être représentée par des séquences d'éléments IGCF qui décrivent le processus de définition. Une telle définition peut être faite en instanciant les objets de niveau inférieur avec des paramètres de transformation spécifiques. Donc, un IGCF doit incorporer des parenthèses pour marquer le début et la fin d'une définition d'objet, d'une instanciation d'objet, et d'une redéfinition d'objet.

Pour compléter le mécanisme de définition d'objet, un IGCF doit permettre l'utilisation d'un alphabet souple pour créer des identifiants d'objet qui assurent l'unicité d'un identifiant dans une hiérarchie. La construction des identifiants d'objet ne fait pas partie d'un IGCF, et l'IGCF a seulement à représenter les identifiants. De plus, un identifiant doit être indépendant d'une session de communication et d'un système graphique particulier afin que les identifiants créés chez un hôte durant une session puissent être utilisés dans d'autres sessions impliquant éventuellement d'autres hôtes, pour rappeler les objets qu'ils marquent.

On laisse aussi aux systèmes en communication la mise en œuvre des mécanismes de résolution des identifiants dupliqués lors de la fusion de deux hiérarchies, créés dans des sessions différentes. Dans cet article, on se limitera à l'avertissement que les numéros de segment ne sont pas qualifiés comme identifiants, parce qu'ils dépendent de la session et de l'état du système dans lequel ils sont créés.

En plus de la définition et de l'instanciation d'objet, un IGCF devrait avoir des éléments représentant les opérations sur les objets. Les opérations identifiées jusqu'à présent sont des transformations, des suppressions, des affichages, des disparitions, des expositions, et des masquages. L'exposition est utilisée pour découvrir sur un écran des objets qui sont cachés par d'autres objets ; le masquage est utilisé pour placer un objet derrière d'autres sur un écran.

4. Proposition d'IGCF

4.1 Fondé sur l'utilisation de GKSM

Un IGCF doit être utilisable pour transmettre toutes les actions graphiques dans une session de conférence. Cela suggère de fonder un IGCF sur un métafichier standard graphique de capture de session, assurant ainsi la compatibilité avec une grande population d'utilisateurs. On a fondé l'IGCF proposé, PIGCF, sur la spécification de métafichier GKSM de capture de session parce que GKSM contient de nombreux éléments identifiés pour un IGCF [14]. De plus, l'orientation vers l'audit de GKSM permet l'enregistrement des sessions interactives de communication pour une exécution ultérieure, et c'est une caractéristique dont on prévoit qu'elle sera fréquemment utilisée.

Le GKSM est un sous-ensemble approprié de notre PIGCF et donc tout système graphique développé pour traiter le PIGCF peut lire un métafichier GKSM. À l'inverse, les applications qui utilisent le PIGCF devraient avoir une option pour restreindre l'enregistrement de session à la seule partie GKSM, éventuellement en supprimant certains événements de session. En faisant ainsi, on sera capable d'acheminer un métafichier GKSM à tout correspondant qui a le logiciel d'interprétation de GKSM. Autrement, une application avec un interpréteur GKSM, mais sans interpréteur PIGCF, peut lire un fichier PIGCF en interprétant seulement la partie GKSM et en ignorant le reste.

Bien que le GKSM ait été spécifié pour le système GKS, on pense que le GKSM est une base raisonnable et générale pour toutes nos applications en deux dimensions. On pense que la spécification GKSM ne se restreint pas aux systèmes GKS mais contient tous les éléments les plus utiles souhaitables dans un métafichier. À l'avenir, on prévoit de s'attaquer aux applications qui requièrent la 3-D, comme les aides interactives à la réparation et la maintenance. Lorsque GKS sera agrémenté de capacités 3-D [13], nous étendrons le PIGCF avec tous les éléments nécessaires.

Nous sommes conscients que la spécification GKSM ne fait pas partie de la norme GKS elle-même mais en est un appendice qui recommande un tel format de métafichier. Néanmoins, toutes les mises en œuvre connues de GKS prennent en charge, pour l'instant la sortie et l'interprétation de métafichier GKSM. Si cette tendance se continue, comme on le

pense, nous serons capables d'échanger des fichiers graphiques avec une large base d'installations GKS. Il y en aura bien sûr beaucoup dans un proche avenir car GKS va être adopté comme norme par l'ISO et par de nombreux organismes nationaux de normalisation.

4.2 Coordonnées des informations de position

Suivant la convention GKSM, les informations de position PIGCF sont en coordonnées normalisées d'appareil, (NDC, *Normalized Device Coordinates*). Donc, l'origine d'une conférence doit indiquer la fenêtre de station de travail pour la conférence. Cette fenêtre est le sous rectangle de l'espace NDC qui enclôt les zones intéressantes pour la conférence. Dans la plupart des cas, les stations de travail participantes vont prendre cette fenêtre à leur compte. Cependant, les systèmes graphiques devraient prévoir la possibilité qu'une station de travail choisisse une fenêtre différente, qui peut contenir la fenêtre de la conférence ou juste la chevaucher. Sauf dans des cas particuliers, une origine de conférence ne devrait pas fixer un angle de vue pour les stations de travail d'une conférence. De cette manière, chaque station de travail peut afficher son angle de vue dans la portion la plus convenable de l'écran.

Il y aura des conférences où les stations de travail participantes vont conserver les informations de position en coordonnées mondiales (WC, *World Coordinates*). Il peut être nécessaire de reconstruire les dimensions mondiales après transmission parce que de telles dimensions ont une signification pertinente pour l'application, comme les tailles des composants ou les distances. Dans ce cas, une station de travail aura à transposer les WC en NDC avant de transmettre et de NDC en WC après réception. À la sortie, l'origine de la conférence devra spécifier la fenêtre mondiale et l'angle de vue NDC utilisé dans la conférence afin que les stations de travail de la conférence fassent de telles transpositions. Ces transpositions devraient être faites par la couche de présentation, selon les termes du modèle de référence d'interconnexion des systèmes ouverts de l'ISO, d'une manière transparente pour les programmes d'application communicants.

Le plus souvent, toutes les stations de travail auront les mêmes fenêtres mondiales et les mêmes angles de vue NDC. Cependant, les systèmes graphiques vont donner la possibilité aux stations de travail de choisir une fenêtre ou un angle de vue différent, mais de telles stations de travail devront enregistrer celui de la conférence pour faire les transpositions susmentionnées. Il y a des systèmes graphiques, comme ACM Core, qui n'assurent pas la transformation de la station de travail. Dans de tels systèmes, l'angle de vue NDC est considéré comme étant la fenêtre de la station de travail pour les transpositions susmentionnées.

4.3 Couches du PIGCF

Il y a deux niveaux dans le PIGCF, un niveau inférieur L (*lower*) et un niveau supérieur U (*upper*). Le niveau inférieur L est juste la spécification du méta fichier GKSM comme défini à l'Appendice E de la proposition de norme ANSI GKS [14]. On a cité un extrait de la plus grande partie de l'Appendice E de [14] à la fin de la présente RFC dans notre Appendice A. Au niveau L, les éléments appartiennent à la mise à jour Group-1 sauf : Établir l'état Différé, les éléments d'attribut de primitive de sortie, les éléments d'attribut de station de travail, Rectangle de découpage, Créer le segment, Fermer le segment, Renommer le segment, Établir la priorité du segment, et Établir la détectabilité.

Le niveau supérieur U est constitué des éléments dont on pense qu'ils complètent le GKSM pour les échanges graphiques en ligne généraux. Cette mise en couche se conforme à la structure de niveau métafichier graphique décrite dans Enderle et autres [15]. Dans une telle structure, un métafichier en mode application peut se fonder sur des métafichiers graphiques.

4.4 Éléments PIGCF dans le niveau U

Les éléments du niveau U sont codés comme des éléments d'utilisateur GKSM de sorte qu'un fichier PIGCF va se conformer à la spécification de métafichier GKSM. En conséquence, un fichier PIGCF sera un métafichier GKSM dans sa totalité. On utilise les mêmes conventions de formatage que dans la spécification GKSM. Ceux qui ne sont pas familiers avec ces conventions devraient lire de début de l'appendice. Les éléments suivants appartiennent au second groupe de mise à jour : les deux éléments pour la définition d'objet, les deux éléments pour la redéfinition d'objet, les deux éléments pour l'instanciation d'objet, les deux éléments pour la transformation de normalisation, Choisir le composant, et Rappeler la bibliothèque. Les éléments restants appartiennent au premier groupe de mise à jour.

Éléments pour la définition d'objet

Début de définition : | 'GKSM 120' | L |

Indique le début de la séquence de définition d'objet

Fin de définition : | 'GKSM 121' | L | I |

Indique la fin de la séquence de définition d'objet.

I(Nc) : identifiant d'objet (N précédant c, i, r signifie un nombre arbitraire de caractères, entiers (*integers*), ou réels.) Les

objets définis de façon interactive sont rendus visibles à l'écran ; c'est-à-dire, ils sont automatiquement instanciés. Si seule la définition doit être gardée, mais pas l'image, un élément Disparaît doit suivre.

Début de redéfinition : | 'GKSM 122' | L | I |
Indique le début de la séquence de définition d'objet.
I(Nc) : identifiant d'objet

Fin de redéfinition : | 'GKSM 123' | L |
Indique la fin de la séquence de redéfinition d'objet

Éléments pour l'instanciation d'objet

Début d'instanciation : | 'GKSM 124' | L | I |
Indique le début de la séquence d'instanciation d'objet
I(Nc) : Identifiant d'objet

Fin d'instanciation : | 'GKSM 125' | L |
Indique la fin de la séquence d'instanciation d'objet

Éléments pour la manipulation d'objet

Transformation d'objet : | 'GKSM 126' | L | C | I | M |
Applique la transformation M à l'objet I
C : nombre de caractères dans l'identifiant
I(Nc) : identifiant d'objet
M(6r) : rangées supérieure et centrale d'une matrice 3x3 représentant une transformation 2D homogène [12].
M 11 M 12 M 13 M 21 M 22 M 23

Suppression d'objet : | 'GKSM 127' | L | I |
I(Nc) : identifiant d'objet

Affichage d'objet : | 'GKSM 128' | L | I |
Rend visible l'objet I
I(Nc) : identifiant d'objet

Disparition d'objet : | 'GKSM 129' | L | I |
Supprime la visibilité de l'objet I
I(Nc) : identifiant d'objet

Expose l'objet : | 'GKSM 130' | L | I |
Affiche l'objet I par dessus tous objets chevauchants
I(c) : identifiant d'objet

Cache l'objet : | 'GKSM 131' | L | I |
Affiche l'objet I derrière tous les objets chevauchants
I(c) : identifiant d'objet

Choix de composant : | 'GKSM 132' | L | I | P |
Choisit le composant P de l'objet I
I(c) : identifiant d'objet
P(i) : prend l'identifiant du composant
Ceci est utilisé pour choisir un groupe de primitives de sortie identifiées par P dans un segment associé à I.

Écraser le composant : | 'GKSM 133' | L | I | P |
Écrase le composant P de l'objet I
I(c) : identifiant d'objet
P(i) : prend l'identifiant du composant
Cela écrase un groupe de primitives de sortie identifiées par P dans un segment associé à I. Cet élément ne peut être utilisé qu'au sein d'une séquence REDÉFINIR L'OBJET.

Éléments pour transformation de normalisation

Établir la fenetre : | 'GKSM 134' | L | W |

Définit les limites de la fenêtre mondiale pour la transformation de normalisation.

W(4r) : limites de la fenêtre mondiale (XMIN, XMAX, YMIN, YMAX)

Établir l'angle de vue : | 'GKSM 135' | L | V |

Définit les limites de l'angle de vue NDC pour la transformation de normalisation.

V(4r) : limites de l'angle de vue NDC (XMIN, XMAX, YMIN, YMAX)

Éléments pour d'autres opérations

Interrompre : | 'GKSM 136' | L |

Interrompt l'opération en cours de transmission dans le flux PIGCF. Cela donne le moyen d'interrompre des opérations non voulues ou erronées. Seule l'opération la plus interne d'une séquence incorporée est interrompue ; des interruptions successives peuvent être utilisées pour sortir de plusieurs niveaux d'incorporation d'opérations.

Suivi de pointeur : | 'GKSM 137' | L | T | P |

Met à jour la position du pointeur graphique sur PT(i) : 0 : cause seulement le déplacement du curseur

1 : cause le traçage du mouvement du curseur par une ligne

P(p) : un point échantillonné à partir de la trace du mouvement du pointeur graphique

Bande réfléchissante : | 'GKSM 138' | L | T | P |

Fait écho à une bande réfléchissante de type T avec des points de référence et de retour. La première occurrence de cet élément dans une séquence porte les coordonnées du point de référence d'écho. Les occurrences suivantes portent les mises à jour d'une position de pointeur qui indiquent un point de retour d'écho.

T(i) : type d'écho (0 : point de référence d'écho ;

> 0 : retour d'écho : 1 = ligne,
2 = rectangle,
3 = cercle)

P(r) : point de référence d'écho (T = 0), ou point de retour d'écho (T > 0)

Les points de référence et de retour sont :

T = 1 – la référence est une extrémité de ligne, le retour est l'autre extrémité.

T = 2 – la référence est un coin de rectangle, le retour est le coin opposé.

T = 3 – la référence est le centre du cercle, le retour est un point du périmètre.

Rappel de bibliothèque : | 'GKSM 139' | L | F |

Rappelle la bibliothèque graphique dans le fichier F

F(i) : nom du fichier qui contient la bibliothèque

Les images graphiques dans F et tous leurs composants deviennent disponibles à l'utilisation durant la session de communication. Les images sont supposées être enregistrées avec le PIGCF, et leurs composants sont à afficher avec les éléments Afficher l'objet ou des actions similaires afin que les images deviennent visibles.

5. Architecture pour le traitement de PIGCF

Cette section présente un exemple d'architecture de logiciel pour la génération et l'interprétation de PIGCF dans un système de conférence multimédia utilisant GKS comme interface graphique du programme sous-jacent. Cette section ne devrait pas être interprétée comme une déclaration définitive d'une telle architecture, mais seulement comme un exercice pour illustrer comment le format proposé dans ce mémoire tient dans le cadre global d'un système de conférence. Choisir GKS simplifie l'exemple d'architecture ; néanmoins, d'autres paquetages graphiques peuvent être utilisés en ajoutant à l'architecture les modules pour interpréter et générer les éléments du niveau L de PIGCF.

La Figure 1 montre les modules logiciels majeurs qui sont chargés de l'interaction et de l'affichage graphique à la station de travail de conférence. C'est une vue de l'écoulement graphique qui est familière au programmeur. Un programme d'application de conférence met à jour les structures de données et utilise des services graphiques indépendants de l'appareil à travers des liens de langage. Ces services, à leur tour, utilisent des services graphiques dépendants de l'appareil qui commandent aux pilotes d'appareil d'accepter les entrées et de présenter les images graphiques. L'application effectue de nombreuses autres fonctions pour la gestion de la conférence et le contrôle des autres flux de support, mais on n'en a pas besoin dans cet exemple.

Dans la Figure 2, le tuyau graphique de base a été augmenté des modules logiciels impliqués dans la génération, la

transmission, la réception, et l'interprétation des flux PIGCF. L'application a un module pour interpréter les niveaux supérieurs et inférieurs de PIGCF et un module pour générer le niveau supérieur U. Les services graphiques indépendants de l'appareil incluent des modules pour générer et interpréter le niveau inférieur L. Cela reflète la pratique actuelle d'inclure les fonctions de génération et d'interprétation dans le paquetage graphique. Il y a aussi un module qui transmet les flux PIGCF sortants aux stations de travail distantes. De même, il y a un module qui reçoit les flux entrants provenant des stations distantes. Dans la pratique actuelle, les modules d'émission et de réception sont décomposés en plusieurs processus qui mettent en œuvre une architecture de protocole en couches. Un processus reçoit les deux niveaux de PIGCF et les écrit dans un métafichier d'enregistrement de conférence pour utilisation ultérieure. Un processus routeur reçoit et transmet le trafic PIGCF de et vers les modules mentionnés précédemment. Ce routeur sera vraisemblablement remplacé par des interfaces de communication indépendantes entre les paires de modules qui échangent du PIGCF.

Les flèches épaisses montrent le flux de PIGCF sortant, tandis que les flèches fines montrent le flux de PIGCF entrant. On suit d'abord le chemin sortant, commençant à l'application. L'application traite les actions d'utilisateur local qui sont transformées en mises à jour de structure de données, en éléments PIGCF de niveau U, et en exécutions de sous-programmes graphiques indépendants de l'appareil qui entre autres choses, génèrent des éléments PIGCF de niveau L (GKSM).

Le routeur fusionne les deux niveaux de flux conformément à l'ordre de génération et les envoie à la copie locale de l'enregistrement de conférence et au module de transmission. Ce dernier entasse les éléments PIGF de groupe 2 jusqu'à ce qu'il reçoive un élément de groupe 1. Il met aussi un horodatage au flux PIGCF pour synchroniser son exécution, chez le receveur, avec l'exécution des autres informations du support. Le PIGCF peut être séparé en catégories de trafic transmises sur diverses facilités de communication selon les services de transport requis par les catégories, par exemple, service en temps réel pour les mises à jour de pointeur, transmission à haute fiabilité pour les définitions de nouveaux objets, ou service à faible priorité pour les transferts de bibliothèque graphique. Finalement, le module de transmission doit accuser réception du PIGCF entrant, et aussi du trafic des autres supports.

Le module receveur est le point d'entrée pour les flux PIGCF entrants qui peuvent venir dans des catégories de trafic diverses qu'il faut fusionner. Il vérifie les horodatages pour la synchronisation des éléments PIGCF qui ont des données qui se rapportent à d'autres supports, par exemple, la voix. Il est possible d'inclure ici une fonction de correction d'erreur de haut niveau qui valide les flux reçus en utilisant les informations d'état et de contexte sur la syntaxe et la sémantique PIGCF. Le module receveur passe les flux au routeur qui les transmet aux trois processus : il envoie les éléments de niveau L à l'interpréteur GKSM qui produit les changements correspondants sur l'image affichée ; il envoie les éléments de niveau L et U à l'enregistrement de conférence, ainsi qu'au code d'interprétation PIGCF dans l'application. Les éléments de niveau U causent la mise à jour des hiérarchies d'objet de modélisation de structures de données ainsi que de la représentation picturale des hiérarchies, à travers l'exécution des services graphiques. Les éléments U mettent aussi à jour les curseurs graphiques et peuvent rappeler de nouvelles bibliothèques graphiques. L'application doit traiter les éléments de niveau L parce qu'ils pourraient indiquer des mises à jour des structures de données ; cela va arriver si, par exemple, la structure enregistre des informations de valeur d'attribut pour les hiérarchies d'objet. L'application coordonne ces actions avec des effets d'autres supports conformément aux horodatages. L'exécution de l'enregistrement de conférence est effectuée en mode hors ligne. Les éléments d'enregistrement sont reçus par le routeur et ensuite traités de façon similaire à celle du PIGCF entrant.

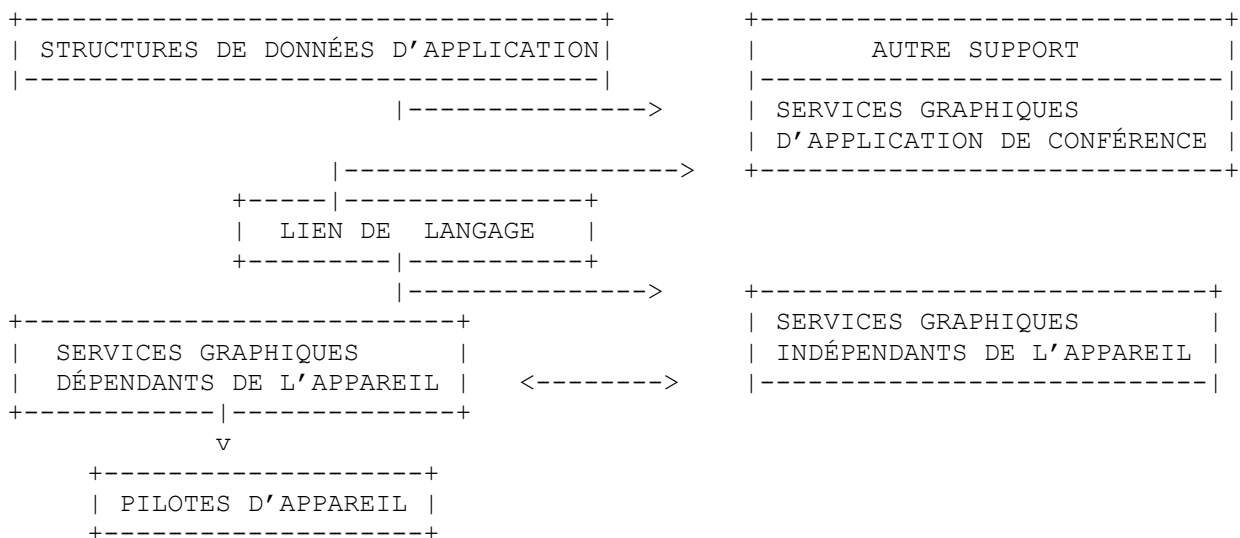


Figure 1 :-Tuyau graphique de base dans un système de conférence

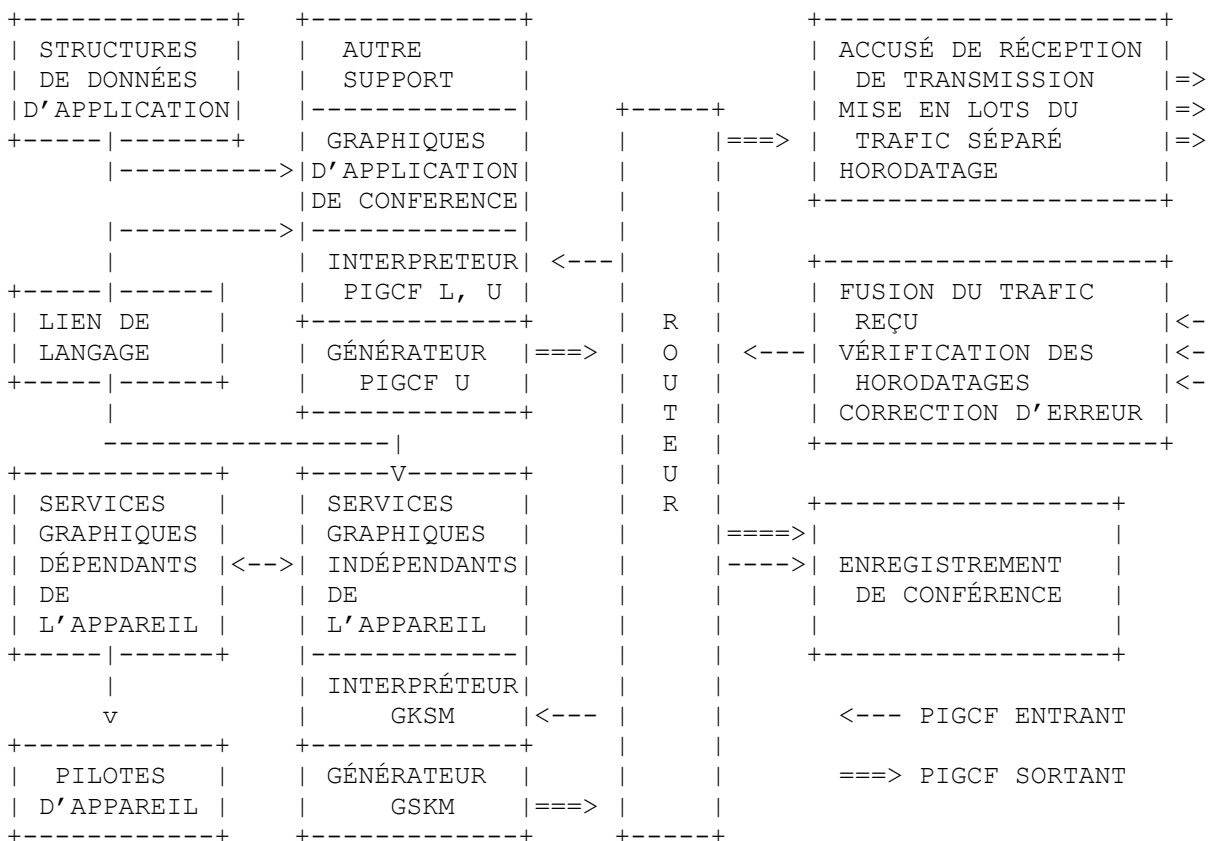


Figure 2 : Architecture de logiciel de conférence pour le traitement du PIGCF

6. Conclusions

Les applications de téléconférence et d'autres multimédia feront partie des ressources de communication disponibles aux entreprises dans un avenir proche. Cela invitera les pratiquants de l'infographie et de la communication informatique à traiter la question de la communication graphique d'application à application. Un protocole est un élément clé de la question, et le format des données est un composant clé du protocole. On a présenté les exigences opérationnelles d'un tel protocole et on a proposé un format qui satisfait à ces exigences.

À présent, aucun des standard graphiques émergents ou existants ne peut être utilisé comme le protocole dont on a besoin ou comme un format pour le protocole, mais cela peut changer car les standard évoluent. On surveille le développement des standard et on étudie l'utilisation de certains d'entre eux comme base de format, en particulier le CGI. Néanmoins, la communauté de la communication informatique a cruellement besoin d'expérience de la mise en œuvre de conférences multimédia. Pour que ces applications se fassent jour, on peut fonder un protocole de communication graphique sur une norme officielle ou un standard de facto qui va vraisemblablement avoir une large utilisation, assurant ainsi l'interopérabilité avec une large base d'utilisateurs. On estime qu'en utilisant le métafichier de session GKSM, on va dans la bonne direction.

La prévision de l'architecture logicielle pour générer et interpréter le PIGCF proposé a posé certains problèmes auxquels on sera confronté lors de la poursuite du travail sur le développement d'un protocole graphique complet. Cela est fait au titre du programme courant de SRI dans les communications multimédia. Dans ce programme, on a mis en œuvre un prototype simple de conférence multimédia et on va en concevoir un plus complet. L'expérience retirée de ces deux exercices sera un apport précieux à la conception de l'architecture du protocole.

Appendice A Extraits de "Projet de proposition de système de noyau graphique" [14]

E.2 Métafichier fondé sur le DIS7942 de l'ISO

Ce métafichier peut être catégorisé comme visant à fournir un moyen d'enregistrer la séquence exacte des invocations de fonctions faites à GKS. Sa capacité fonctionnelle couvre la gamme entière des fonctions de sortie de GKS, du niveau m au niveau 2. Il convient donc pour les applications où les actions graphiques individuelles doivent être exécutées peut-être avec une édition graphique sélective effectuée par l'interpréteur.

Deux codages ont été spécifiés pour ce métafichier. Un codage est inefficace pour de nombreuses applications. Le second permet un format binaire non spécifié. Le reste de cet appendice sur IGCP donne tous les détails de ces structures et codages de métafichier.

E.2.1 Format de fichier et de données

Le métafichier GKS est construit comme une séquence d'éléments de données logiques. Le fichier commence par un en-tête de fichier de format fixe qui décrit l'origine du métafichier (auteur, installation) le format des éléments suivants, et le nombre de représentations. Le fichier se termine par un élément terminal qui indique la fin logique du fichier. Entre ces deux éléments, les informations suivantes sont enregistrées dans le sens d'un chemin d'examen :

- a) éléments de contrôle de la station de travail et éléments du message ;
- b) éléments de primitive de sortie, décrivant les objets graphiques élémentaires ;
- c) informations d'attributs, incluant les attributs de primitive de sortie ; les attributs de segments et de station de travail ;
- d) éléments de segments, décrivant la structure de segment et les manipulations dynamiques de segment ;
- e) éléments d'utilisateur.

La structure globale du métafichier GKS est la suivante :

```
FICHER : |en-tête de fichier |élément 1|---|élément i|---|élément final|
ÉLÉMENT : |en-tête d'élément |enregistrement des données d'élément|
ÉLÉMENT : '|GKSM' | identification| longueur des données d'élément|
EN-TÊTE : |facultatif| nombre | en octets |
```

Tous les éléments de données sauf l'en-tête de fichier ont un en-tête d'élément contenant :

- a) la chaîne de caractères 'GKSM' (facultative) qui est présente pour améliorer la lisibilité du fichier et fournir une facilité de contrôle d'erreur ;
- b) le numéro d'identification du type d'élément qui indique la sorte d'informations contenues dans l'élément ;
- c) la longueur de l'enregistrement des données de l'élément.

Les longueurs de ces champs de l'en-tête d'élément dépendent de la mise en œuvre et sont spécifiées dans l'en-tête de fichier.

Le contenu de l'enregistrement des données d'élément est décrit entièrement ci-dessous pour chaque type d'élément.

Le métafichier contient des caractères, des nombres entiers, et des nombres réels marqués (c), (i), (r) dans la description d'élément.

Les caractères dans le métafichier sont représentés conformément aux normes ISO 646 et ISO 2022. Les nombres seront représentés conformément à ISO 6093 en utilisant le format F1 pour les entiers et le format F2 pour les réels. (Remarque que les formats F1 et F2 peuvent être écrits et lus via les formats FORTRAN respectivement I et F.)

Les nombres réels qui décrivent des coordonnées et des unités de longueur sont mémorisés comme des coordonnées d'appareil normalisées. La transformation à la station de travail, si elle est spécifiée dans le programme d'application pour une station de travail qui écrit un métafichier de ce format, n'est pas effectuée mais Fenêtre de station de travail et Accès de vue de station de travail sont mémorisés dans les éléments de données pour utilisation ultérieure. Les nombres réels peuvent être mémorisés comme entiers. Dans ce cas, les paramètres de transformation sont spécifiés dans l'en-tête de fichier pour permettre une transformation appropriée des entiers en coordonnées d'appareil normalisées.

Pour des raisons d'économie, les nombres peuvent être mémorisés en utilisant un format binaire interne. Comme il n'existe pas de norme pour la représentation des nombres binaires, ce format limite la portabilité du métafichier. La spécification d'une telle représentation de nombre binaire sort du domaine d'application du présent document.

Lors de l'échange de métafichiers entre différentes installations, la structure physique des ensembles de données sur un support de mémorisation spécifique devrait être normalisée. Une telle définition sort du domaine d'application de ce standard.

E.3 Génération des métafichiers

Le Tableau E1 contient une liste, par classe, de toutes les fonctions GKS qui s'appliquent aux stations de travail de catégorie MO, et de leurs effets sur ce GKSM. Dans le tableau, GKSM-OUT est un identifiant de station de travail qui indique une station qui écrit un métafichier de ce format.

Les concepts de rectangle de découpage et d'indicateur de découpage sont encapsulés dans un élément de métafichier qui spécifie un rectangle de découpage. Cet élément est écrit sur le métafichier de la station de travail active avec les valeurs (0, 1, 0, 1), si l'indicateur de découpage est fermé, ou sur l'accès de vue de la transformation de normalisation en cours, si l'indicateur de découpage est ouvert. Si l'accès de vue de la transformation de normalisation en cours est redéfini ou si une transformation de normalisation différente est choisie lorsque l'indicateur de découpage est ouvert, un autre élément de rectangle de découpage est écrit. Si l'indicateur de découpage est passé à fermé, un élément de rectangle de découpage (0, 1, 0, 1) est écrit. Si l'indicateur de découpage est passé à ouvert, un élément contenant l'accès de vue de la transformation de normalisation en cours est écrit. Ceci est analogue au traitement du découpage dans les segments (voir au paragraphe 4.7.6 de [14]).

Fonctions GKS qui s'appliquent aux stations de travail de catégorie MO	Élément ou effet GKSM créé
Fonctions de contrôle	
Ouvrir la station de travail (GKSM-OUT,...)	- (en-tête de fichier) 1 (CONDITIONNEL)
Fermer la station de travail (GKSM-OUT) travail (GKSM-OUT)	0 (élément final) Activer la station de (61, 21-44) assure les attributs courants ; permet la sortie
Désactiver la station de travail (GKSM-OUT)	désactive la sortie
Supprimer la station de travail (GKSM-OUT,...)	1 2
Redessiner tous les segments sur la station (GKSM-OUT)	
Mettre à jour la station (GKSM-OUT,...)	3
Établir l'état d'attente (GKSM-OUT,...)	4
Message (GKSM-OUT,...)	5 (message)
Échappement	6
<hr/>	
Primitives de sortie	
Polyligne	11
Polymarqueur	12
Texte	13
Zone de remplissage	14
Dispositif en cellules	15
Primitive de dessin généralisée	16
<hr/>	
Attributs de sortie	
Établir l'indice de polyligne	21
Établir le type de ligne	22
Établir le facteur d'échelle de largeur de ligne	23
Établir l'indice de couleur de polyligne	24
Établir l'indice de polymarqueur	25
Établir le type de marqueur	26
Établir le facteur d'échelle de taille de marqueur	27
Établir l'indice de couleur de polymarqueur	28
Établir l'indice de texte	29
Établir la font et la précision du texte	30
Établir le facteur d'expansion de caractère	31
Établir l'espacement de caractère	32
Établir l'indice de couleur de texte	33
Établir la hauteur de caractère	34
Établir le vecteur de caractère montant	34
Établir le chemin de texte	35
Établir l'alignement de texte	36
Établir l'indice de zone de remplissage	37
Établir le style intérieur de zone de remplissage	38
Établir l'indice de style de zone de remplissage	39
Établir l'indice de couleur de zone de remplissage	40
Établir la taille du motif	41

Établir un point de référence de motif	42
Établir les fanions de source d'aspect	43
Établir l'identifiant de pic	44

Attributs de station de travail

Établir la représentation de polyligne (GKSM-OUT,...)	51
Établir la représentation de polymarqueur (GKSM-OUT,...)	52
Établir la représentation de texte (GKSM-OUT,...)	53
Établir la représentation de zone de remplissage (GKSM-OUT,...)	54
Établir la représentation de schéma (GKSM-OUT,...)	55
Établir la représentation de couleur (GKSM-OUT,...)	56

Fonctions de transformation

Établir la fenêtre de la transformation de normalisation en cours	34, 41, 42
Établir le point de vue de la transformation de normalisation en cours	61, 34, 41, 42
Choisir la transformation de normalisation en cours	61, 34, 41, 42
Établir l'indicateur de découpage	61
Établir la fenêtre de la station de travail (GKSM-OUT,...)	71
Établir le point de vue de la fenêtre de la station de travail (GKSM-OUT,...)	72

Note : l'élément 61 (Rectangle de découpage) est décrit plus complètement en E.2.2.

Note : Lorsque la transformation de normalisation en cours est altérée, les éléments correspondant aux attributs qui contiennent les informations de coordonnées sont envoyés (éléments 34, 41, et 42).

Fonctions de segment

Créer le segment	81
Fermer le segment	82
Renommer le segment	83
Supprimer le segment	84
Supprimer le segment de la station de travail (GKSM-OUT,...)	84
Associer le segment à la station de travail (GKSM-OUT,...)	81, (21-44), (11-16), (61), 82
Copier le segment sur la station de travail (GKSM-OUT,...)	(21-44), (11-16), (61)
Insérer le segment	(21-44), (11-16), (61)

Attributs de segment

Établir la transformation de segment	91
Établir la visibilité	92
Établir le surlignement	93
Établir la priorité de segment	94
Établir la détectabilité	95

Fonctions de métafichier

Écrire l'élément en GKSM	> 100
--------------------------	-------

E.4 Interprétation des métafichiers

E.4.1 Introduction

L'interprétation des métafichiers en GKS est décrite au paragraphe 4.9 de [14]. Les effets de Interpréter l'élément pour tous les types d'élément de métafichier sont décrits dans les paragraphes suivants. Les éléments sont groupés par classe de fonction.

E.4.2 Éléments de contrôle

L'interprétation des éléments de cette classe est décrite sous la définition de chaque élément en E.5. ([14] dit "E.2.4" au lieu de "E.5" ce qui est une erreur).

E.4.3 Primitives de sortie

L'interprétation des éléments de cette classe génère un résultat correspondant aux fonctions de primitive, sauf que les coordonnées des points sont exprimées en NDC. Les attributs de primitive liés aux primitives sont ceux qui ont été générés par l'interprétation des éléments d'attribut de primitives dans ce métafichier particulier (voir en E.4.4).

E.4.4 Attributs des primitives de sortie

L'interprétation des éléments de cette classe établit des valeurs à utiliser pour l'affichage des primitives générées ensuite à partir de ce métafichier particulier (voir en E.4.3). Aucun changement n'est fait aux entrées dans la liste d'états GKS.

E.4.5 Attributs de station de travail

L'interprétation des éléments de cette classe a le même effet que l'invocation de la fonction GKS correspondante montrée au Tableau E1. Les fonctions GKS sont effectuées sur toutes les stations de travail actives.

E.4.6 Transformations

L'interprétation d'un élément rectangle de découpage établit des valeurs à utiliser dans les primitives de découpage de sortie générées ultérieurement à partir de ce métafichier particulier. Aucun changement n'est fait aux entrées dans la liste des états GKS. L'interprétation des autres éléments de cette classe (Fenêtre de station de travail et Accès de vue de station de travail) cause l'invocation des fonctions GKS correspondantes sur toutes les stations actives.

E.4.7 Manipulation de segment

L'interprétation des éléments de cette classe a le même effet que l'invocation des fonctions GKS correspondantes montrées au Tableau E1. (L'élément 84 cause une invocation de Supprimer le segment.)

E.4.8 Attributs de segment

L'interprétation des éléments de cette classe a le même effet que l'invocation des fonctions GKS correspondantes montrées au Tableau E1.

E.5 Éléments de contrôle

En-tête de fichier : | GKSM | N | D | V | H | T | L | I | R | F | RI | ZÉRO | UN |

Tous les champs de l'élément En-tête de fichier ont une longueur fixe. Les nombres sont formatés conformément au format F1 de la norme ISO 6093.

Informations générales :

GKSM : 4 octets ; contient la chaîne 'GKSM'

N : 40 octets ; contenant le nom de l'auteur ou de l'installation.

D : 8 octets ; date (an/mois/jour, par exemple, 79/12/31).

V : 2 octets ; numéro de version : le métafichier décrit ici a le numéro de version 1.

H : 2 octets ; entier spécifiant combien d'octets de la chaîne 'GKSM' sont répétés au début de chaque enregistrement.
Les valeurs possibles sont : 0, 1, 2, 3, 4.

T : 2 octets ; longueur d'un champ d'indicateur de type d'élément.

L : 2 octets ; longueur du champ d'indicateur de longueur d'enregistrement de données d'élément.

I : 2 octets ; longueur du champ pour chaque entier dans l'enregistrement de données d'élément (appliqué à toutes les données marquées (i) dans la description d'élément).

R : 2 octets ; longueur du champ pour chaque nombre réel dans l'enregistrement de données d'élément (s'applique à toutes les données marquées (r) dans la description d'élément).

Spécification de la représentation de nombre :

F : 2 octets ; valeurs possibles : 1, 2. Cela s'applique à toutes les données dans les éléments marqués (i) ou (r) et à la longueur d'enregistrement de type d'élément et de données d'élément :

1 : tous les nombres sont formatés conformément à ISO 6093,

2 : tous les nombres (sauf dans l'en-tête de fichier) sont mémorisés dans un format binaire interne.

RI : 2 octets ; valeurs possibles : 1, 2. Représentation de nombre pour les données marquées (r) : 1 = réel, 2 = entier.

ZÉRO : 11 octets ; entier équivalent à 0.0, si RI = 2

UN : 11 octets ; entier équivalent à 1.0, si RI = 2

Après l'en-tête de fichier, qui est de format fixe, toutes les valeurs dans les éléments suivants sont dans le format défini par l'en-tête de fichier. Pour la description qui suit, on suppose le réglage suivant : H = 4; T = 3; F = 1. En plus des formats (c), (i) et (r), qui sont déjà décrits, (p) note un point représenté par une paire de nombres réels (2r). Cette notation permet de faire précéder une seule lettre par une expression, qui indique le nombre de valeurs de ce type.

{Des commentaires explicatifs ont été ajoutés à certaines spécifications d'élément ; elles ne font pas partie de l'Appendice E à GKS et sont incluses entre des accolades {} . Une définition complète de la génération et de l'interprétation des éléments GKSM est donnée par la définition des fonctions GKS correspondantes dans [14].}

Élément final : | 'GKSM 0' | L |

Dernier élément de chaque métafichier GKS. Établit la condition de l'erreur.

Supprimer la station de travail : | 'GKSM 1' | L | C |

Demande la suppression de la station de travail sur toutes les stations de travail actives.

C(i) : suppression du fanion de contrôle ; (0 = Conditionnel, 1 = Toujours)

Redessiner tous les segments sur la station de travail : | 'GKSM 3' | L | R |

Demande la mise à jour de la station sur toutes les stations de travail actives.

R(i) : fanion de régénération : 0 = Effectuer, 1 = Suspendre

État Différer : | 'GKSM 4' | L | D | R |

Demande l'état Différer sur toutes les stations de travail actives.

D(i) : mode Différer : 0 = ASAP, 1 = BNIG, 2 = BNIL, 3 = ASTI

R(i) : mode de régénération implicite : 0 = permis, 1 = supprimé

{Cet élément donne le contrôle de l'occurrence de l'effet visuel des fonctions GKS afin d'optimiser l'utilisation des capacités de la station de travail selon les besoins de l'application.}

Message : | 'GKSM 5' | L | N | T |

Demande Message sur toutes les stations de travail actives.

N(i) : nombre de caractères dans la chaîne.

T(Nc) : chaîne avec N caractères.

{Le message ne fait pas partie des primitives de sortie d'un métafichier ; le message est seulement destiné à être interprété par les opérateurs de station de travail.}

Échappement : | 'GKSM 6' | L | FI | L | M | I | R |

Demande l'échappement

FI(i) : identifiant de fonction

L(i) : longueur de données d'entier dans l'enregistrement de données.

M(i) : longueur des données de réels dans l'enregistrement de données.

I(Li) : données d'entier.

R(Mr) : données de réel.

{Cet élément permet l'invocation d'une fonction d'échappement spécifique non standard FI. L'exécution de la fonction avec les paramètres donnés ne doit pas altérer la liste d'états GKS ni produire de résultat géométrique.}

E.6 Éléments pour les primitives de sortie

Polyligne : | 'GKSM 11' | L | N | P |

N(i) : nombre de points de la polyligne.

P(Np) : liste des points.

Polymarqueur : | 'GKSM 12' | L | N | P |

N(i) : nombre de points.

P(Np) : liste des points.

Texte : | 'GKSM 13' | L | P | N | T |

P(p) : point de départ de la chaîne de caractères.

N(i) : nombre de caractères dans la chaîne T.

T(Nc) : chaîne avec N caractères tirés de l'ensemble de ISO 646

Zone de remplissage : | 'GKSM 14' | L | N | P |

N(i) : nombre de points.

P(Np) : liste des points.

Dispositif en cellules : | 'GKSM 15' | L | P | Q | R | N | M | CT |

P(p),Q(p),R(p) : coordonnées des points des coins de la matrice de pixels (P et Q sont les images des points P et Q spécifiés dans la fonction Dispositif en cellules et R est un autre coin).

M(i) : nombre de rangées dans la matrice.

N(i) : nombre de colonnes dans la matrice.

CT(MNi) : matrice d'indices de couleurs mémorisés rangée par rangée.

{Cet élément permet de passer des images matricielles à GKS. L'image matricielle est définie par la matrice d'indice de couleur CT, et sa position en coordonnées mondiales est donnée par les points P et Q.}

Primitive de dessin généralisée : | 'GKSM 16' | L | GI | N | P | L | M | I | R |

GI(i) : identifiant GDP.

N(i) : nombre de points.

P(Np) : liste des points.

L(i) : longueur des données d'entier dans l'enregistrement de données.

M(i) : longueur des données réelles dans l'enregistrement de données.

I(Li) : données d'entier.

R(Mr) : données de réel.

{Cet élément donne un moyen standard pour dessiner des primitives de sortie supplémentaires non standard. La primitive de dessin généralisée GI est dessinée conformément à la liste P de points et l'enregistrement de données dans I et R.}

E.7 Éléments pour les attributs de primitive de sortie

Indice de polyligne : | 'GKSM 21' | L | LT |

LT(i) : type de ligne.

Facteur d'échelle de largeur de ligne : | 'GKSM 23' | L | LW |

LW(r) : facteur d'échelle de largeur de ligne.

{Dans GKS, la largeur de ligne n'est pas affectée par les transformations GKS. Cependant, la largeur effective de ligne est calculée comme le produit de la largeur nominale de ligne par le facteur d'échelle utilisé lors du tracé de la ligne.}

Indice de couleur de polyligne : | 'GKSM 24' | L | CI |

CI(i) : indice de couleur de polyligne.

Indice de polymarqueur : | 'GKSM 25' | L | I |

I(i) : indice de polymarqueur.

Type de marqueur : | 'GKSM 26' | L | MT |

MT(i) : type de marqueur

Facteur d'échelle de taille de marqueur : | 'GKSM 27' | L | MS |

MS(r) : facteur d'échelle de taille de marqueur.

{Dans GKS, la taille du marqueur n'est pas affectée par les transformations GKS. Cependant, la taille effective du marqueur est calculée comme le produit de la taille nominale du marqueur par le facteur d'échelle de taille du marqueur utilisée lors du traçage d'un marqueur.}

Indice de couleur de polymarqueur : | 'GKSM 28' | L | CI |

CI(i) : indice de couleur de polymarqueur.

Indice de texte : | 'GKSM 29' | L | I |

I(i) : indice de texte.

Fonte et précision de texte : | 'GKSM 30' | L | F | P |

F(i) : fonte du texte.

P(i) : précision du texte. (0 = Chaîne, 1 = Caractère, 2 = Trait)

Facteur d'expansion de caractère : | 'GKSM 31' | L | CEF |

CEF(r) : facteur d'expansion de caractère.

{Cet élément permet la manipulation de la largeur/hauteur du corps de caractère. La largeur du corps de caractère est réglée par le facteur CEF.}

Espacement de caractère : | 'GKSM 32' | L | CS |

CS(r) : espacement de caractère.

Indice de couleur de texte : | 'GKSM 33' | L | CI |

CI(i) : indice de couleur de texte.

Vecteurs de caractères : | 'GKSM 34' | L | CH | CW |

CH(2r) : vecteur de hauteur de caractère.

CW(2r) : vecteur de largeur de caractère.

Note : Ces vecteurs sont les vecteur de hauteur et de largeur décrits au paragraphe 4.4.5 de [14].

{Le vecteur de hauteur de caractère est parallèle au vecteur de caractère montant et a une longueur égale à la hauteur du caractère. La hauteur de caractère spécifie la hauteur d'une lettre majuscule. Le vecteur de largeur de caractère est perpendiculaire au vecteur de hauteur, dans la direction de la ligne de base des caractères, et a la même longueur.}

Chemin de texte : | 'GKSM 35' | L | P |

P(i) : chemin de texte. (0 = Gauche, 1 = Droite, 2 = Montant, 3 = Descendant)

Alignement du texte : | 'GKSM 36' | L | H | V |

H(i) : alignement horizontal des caractères ; (0 = Normal, 1 = Gauche, 2 = Centré, 3 = Droit)

V(i) : alignement vertical des caractères ; (0 = Normal, 1 = Sommet, 2 = Chapeau, 3 = Demi, 4 = Base, 5 = Bas)

Indice de zone de remplissage : | 'GKSM 37' | L | I |

I(i) : indice de zone de remplissage.

Style intérieur de zone de remplissage : | 'GKSM 38' | L | S |

S(i) : style intérieur de zone de remplissage ; (0 = Trou, 1 = Plein, 2 = Motif, 3 = Hachuré)

Indice de style de zone de remplissage : | 'GKSM 39' | L | SI |

SI(i) : indice de style de zone de remplissage.

Indice de couleur de zone de remplissage : | 'GKSM 40' | L | CI |

CI(i) : indice de couleur de zone de remplissage.

Taille de motif : | 'GKSM 41' | L | PW | PH |

PW(2r) : vecteur de largeur de motif.

PH(2r) : vecteur de hauteur de motif.

{Un style pour les zones de remplissage a un motif de cellules de couleur. Un tel motif est défini par une matrice d'indices de couleurs qui est transposée en un motif rectangulaire dont les dimensions sont données par PW et PH.}

Point de référence de motif : | 'GKSM 42' | L | P |

P(p) : point de référence.

{Un style pour les zones de remplissage a un motif de cellules de couleur. Un tel motif est défini par une matrice d'indices de couleurs qui est transposée en un motif rectangulaire dont le coin gauche est donné par P.}

Fanions de source d'aspect : | 'GKSM 43' | L | F |

F(13i) : fanions de source d'aspect ; (0 = Groupé, 1 = Individuel).

{Une application peut régler un attribut de primitive de sortie à groupé ou à individuel. Les attributs groupés dépendent de la station de travail ; leur liaison est retardée, et leurs valeurs peuvent changer de façon dynamique. Les attributs individuels sont des attributs globaux et ils sont liés immédiatement ; leur valeur est statique et ne peut être manipulée.}

Identifiant de pic : | 'GKSM 44' | L | P |

P(i) : identifiant de pic.

E.8 Éléments pour les attributs de station de travail

Représentation de polyligne : | 'GKSM 51' | L | I | LT | LW | CI |

I(i) : indice de polyligne.

LT(i) : nombre de type de ligne.

LW(r) : facteur d'échelle de largeur de ligne.

CI(i) : indice de couleur de polyligne.

Représentation de polymarqueur : | 'GKSM 52' | L | I | MT | MS | CI |

I(i) : indice de polymarqueur.

MT(i) : type de marqueur.

MS(r) : facteur d'échelle de taille de marqueur.

CI(i) : indice de couleur de polymarqueur.

Représentation de texte : | 'GKSM 53' | L | I | F | P | CEF | CS | CI |

I(i) : indice de texte.

F(i) : fonte du texte.

P(i) : précision du texte. (0 = Chaîne, 1 = Caractère, 2 = Trait).

CEF(r) : facteur d'expansion de caractère.

CS(r) : espacement de caractères.

CI(i) : indice de couleur de texte.

Représentation de zone de remplissage : | 'GKSM 54' | L | I | S | SI | CI |

I(i) : indice de zone de remplissage.

S(i) : style intérieur de zone de remplissage ; (0 = Trou, 1 = Plein, 2 = Motif, 3 = Hachuré)

SI(i) : indice de style de zone de remplissage.

CI(i) : indice de couleur de zone de remplissage.

Représentation de motif : | 'GKSM 55' | L | I | N | M | CT |

I(i) : indice de motif.

N(i) : nombre de colonnes dans la matrice* { * le document ANSI dit "zone" au lieu de "matrice" . }

M(i) : nombre de rangées dans la matrice.

CT(MNi) : tableau des indices de couleur mémorisées par rangée.

{Un style pour les zones de remplissage a un motif de cellules de couleur. Un tel motif est défini par une représentation de motif. }

Représentation de couleurs : | 'GKSM 56' | L | CI | RGB |

CI(i) : indice de couleur.

RGB(3r) : intensités de rouge, vert, bleu.

E.9 Éléments pour les transformations

Rectangle de découpage : | 'GKSM 61' | L | C |

C(4r) : limites du rectangle de découpage (XMIN, XMAX, YMIN, YMAX)

Fenêtre de station de travail : | 'GKSM 71' | L | W |

W(4r) : limites de la fenêtre de la station de travail (XMIN, XMAX, YMIN, YMAX).

{GKS comporte une transformation de station de travail qui transpose un rectangle de l'espace NDC (une fenêtre de station de travail) en un rectangle de l'espace de coordonnées de l'appareil (un accès de vue de station de travail.)}

Accès de vue de station de travail : | 'GKSM 72' | L | V |

V(4r) : limites de l'accès de vue de station de travail (XMIN, XMAX, YMIN, YMAX).

E.10 Éléments pour la manipulation de segment

Créer un segment : | 'GKSM 81' | L | S |

S(i) : nom du segment.

Fermer un segment : | 'GKSM 82' | L |

indique la fin d'un segment.

Renommer un segment : | 'GKSM 83' | L | SO | SN |

SO(i) : ancien nom du segment.

SN(i) : nouveau nom du segment.

Supprimer un segment : | 'GKSM 84' | L | S |

S(i) : nom du segment.

E.11 Éléments pour les attributs de segment

Établir une transformation de segment : | 'GKSM 91' | L | S | M |

S(i) : nom du segment.

M(6r) : transformation des rangées supérieure et centrale d'une matrice 3x3 représentant une transformation 2D homogène
[9] M 11 M 12 M 13 M 21 M 22 M 23

{Cela diffère du document ANSI X3.124 du 5 janvier 1984, par les éléments indiqués de la matrice. On pense qu'il y a une erreur dans ce document. }

Établir la visibilité : | 'GKSM 92' | L | S | V |

S(i) : nom du segment.

V(i) : visibilité ; (0 = Visible, 1 = Invisible)

Établir le surlignage : | 'GKSM 93' | L | S | H |

S(i) : nom du segment.

H(i) : surlignement (0 = NORMAL, 1 = HIGHLIGHTED)

Établir la priorité de segment : | 'GKSM 94' | L | S | P |

S(i) : nom de segment.

P(r) : priorité de segment.

Établir la détectabilité : | 'GKSM 95' | L | S | D |

S(i) : nom du segment.

D(i) : détectabilité ; (0 = Indétectable, 1 = Détectable)

E.12 Éléments d'utilisateur

Élément d'utilisateur : | 'GKSMXXX' | L | D |

XXX : > 100

D : données d'utilisateur (L octets)

{Les éléments PIGCF de niveau U sont codés comme des éléments Utilisateur GKSM de sorte qu'un fichier PIGCF va se conformer à la spécification de métafichier GKSM.}

Appendice B Exemple d'utilisation de PIGCF en conférence

La présente section présente un exemple qui illustre le composant graphique PIGCF proposé dans un échange de conférence audio-graphique. On présente seulement la partie graphique de l'échange de conférence, qui en fait serait complété par de la parole. Par souci de brièveté, l'exemple ne contient pas toute la négociation de paramètres que requerrait l'établissement d'une conférence.

L'exemple porte sur une conférence en ligne audio-graphique entre le commandement de la marine, un centre de contrôle et des bâtiments. Les éléments PIGCF présentés n'appartiennent pas à un seul flux de transmission. Le flux auquel ils appartiennent est déterminé par la station qui les transmet, et l'identification de l'émetteur appartient aux protocoles de communication de niveau inférieur. On utilise le codage des caractères, plutôt que le codage binaire, pour cet exemple de PIGCF. On illustre seulement quelques uns des groupes d'éléments possibles qui pourraient être joints à cet exemple. Le scénario de l'exemple est le suivant.

Le centre de commande (centre) établit une conférence avec certains navires d'une escadre (plates-formes) pour coordonner l'interception d'un vaisseau non identifié aperçu dans une zone de conflit. Après rappel des bibliothèques graphiques, tous les sites de la conférence peuvent voir sur leurs écrans une carte de la zone visée ainsi que des représentations iconiques des navires de l'escadre. Puis le centre dessine de façon interactive une représentation iconique du vaisseau non identifié, le met à l'échelle, et le place dans localisation de visée.

Les plates-formes expliquent les actions possibles en utilisant des pointeurs graphiques. Le centre dessine la trajectoire prévue du vaisseau non identifié et les plates-formes situent les icones de l'escadre aux points d'interception prévus. Puis le centre fait un agrandissement sur la zone d'interception et les plates-formes utilisent des bandes en surbrillance pour exposer les manœuvres d'interception.

On voit ensuite la liste des éléments PIGCF échangés. Le centre initie l'établissement de la conférence graphique avec l'élément En-tête de fichier pour établir les paramètres de base de la représentation pour les informations graphiques à échanger. Cet élément peut être interprété selon sa définition en E.5 [14]. Les choix de paramètres les plus importants pour cet exemple sont :

- i) Les éléments contiennent 0 caractère de la chaîne "GKSM" dans le champ identification de l'en-tête d'élément.
- ii) Le champ Indicateur de type d'élément contenant le nombre d'éléments PIGCF est long de trois octets dans chaque élément.
- iii) Les entiers font 4 octets, et les réels 6 octets.
- iv) L'indicateur de longueur d'enregistrement de données d'élément fait deux octets.

On va respecter les longueurs de champ de la spécification PIGCF et les réglages de longueur de champ susmentionnés. Cependant, on ajoutera une espace avant et après le séparateur "|" pour améliorer la lisibilité. Aussi, chaque élément sera précédé de son nom pour faciliter l'identification.

En-tête de fichier : | GKSM | centre | 84/11/10 | 1 | 0 | 3 | 2 | 4 | 6 | 1 | 1 | | | |

Le centre déclare les limites de la fenêtre de la station de travail pour la conférence.

Fenêtre de station de travail : | 71 | 24 | 0,0 0,5 0,0 0,375 |

Dans cet exemple, on suppose que les stations de travail en conférence utilisent les coordonnées mondiales pour la représentation interne des informations positionnelles. En conséquence, le centre déclare les limites de la fenêtre mondiale

pour la transformation de normalisation utilisée dans la conférence.

Établir la fenêtre : | 134 | 28 | 0,0 320,0 0,0 240,0 |

Le centre informe de la localisation de son accès de vue NDC local, cependant, les autres participants à la conférence peuvent choisir des accès de vue NDC différents pour la même transformation, mais leur fenêtre de station de travail devrait comporter celle de la conférence. Tous les systèmes enregistrent la conférence : fenêtre mondiale, accès de vue NDC, et fenêtre de station de travail.

Établir l'accès de vue : | 135 | 28 | 0,0 0,5 0,0 0,375 |

Le centre rappelle les bibliothèques graphiques qui contiennent les cartes géographiques de la zone de crise et les icônes des escadres dans la zone. Il affiche aussi un objet graphique qui fournit une image d'arrière plan de la zone.

Rappel bibliothèque : | 139 | 9 | caraïbe |
Afficher l'objet : | 128 | 11 | lignes_de_côte |
Rappel bibliothèque : | 139 | 10 | unités_de_l'escadre |

Le centre procède à l'instanciation d'une des escadres de la bibliothèque unités_de_l'escadre. Cela est fait en rappelant certains objets de la bibliothèque et en appliquant ultérieurement des transformations aux objets. Comme Établir la fenêtre, établir l'accès de vue, et rappeler la bibliothèque appartiennent au groupe de mise à jour 2, ils peuvent être mis en attente jusqu'à ce que l'objet affiché, qui est du groupe de mise à jour 1, soit entré. Le second rappel de bibliothèque peut être groupé avec le début d'instanciation suivant jusqu'à ce que soit produit l'objet affiché. Le reste de l'exemple contient plus de cas de séquences d'éléments qui peuvent être groupés ; cependant, pour faire court, on n'en indique pas plus.

Début d'instanciation : | 124 | 15 | US_CONSTITUTION |
Affichage de l'objet : | 128 | 15 | US_CONSTITUTION |
Transformer l'objet : | 126 | 55 | 15 | US_CONSTITUTION | 0,1 0,0 0,0 0,0 0,1 0,0 |
Transformer l'objet : | 126 | 55 | 15 | US_CONSTITUTION | 0,1 0,0 0,312 0,0 0,1 0,078 |
Fin d'instanciation : | 125 | 0 |

Début d'instanciation : | 124 | 13 | US_NEW_JERSEY |
Affichage de l'objet : | 128 | 13 | US_NEW_JERSEY |
Transformer l'objet : | 126 | 53 | 13 | US_NEW_JERSEY | 0,1 0,0 0,0 0,0 0,1 0,0 |
Transformer l'objet : | 126 | 53 | 13 | US_NEW_JERSEY | 0,1 0,0 0,312 0,0 0,1 0,093 |
Fin d'instanciation : | 125 | 0 |

Ensuite, le centre établit les valeurs pour deux attributs de primitive de sortie pour préparer le dessin d'un nouvel icône sur les écrans. On suppose que tous les autres attributs ont été mis à des valeurs par défaut à la suite de l'établissement de la conférence.

Indice de polyligne : | 21 | 4 | 20 |
Indice de couleur de polyligne : | 24 | 4 | 200 |

L'élément suivant correspond à la définition interactive du vaisseau non identifié. Comme la définition est faite de façon interactive, l'image du vaisseau reste visible sur les écrans après la définition.

Commencer la définition : | 120 | 0 |
Polyligne : | 11 | 64 | 5 | 0,047 0,063 0,063 0,047 0,125 0,047 0,14 0,063 0,047 0,047 |
Polyligne : | 11 | 52 | 3 | 0,078 0,063 0,078 0,078 0,109 0,078 0,109 0,063 |
Fin de définition : | 121 | 8 | visée |

Puis, la "visée" du vaisseau non identifié est mise à l'échelle et placée sur le site de visée.

Début instanciation : | 124 | 8 | visée |
Transformer l'objet : | 126 | 48 | 8 | visée | 0,2 0,0 0,0 0,0 0,2 0,0 |
Transformer l'objet : | 126 | 48 | 8 | visée | 0,1 0,0 0,156 0,0 0,1 0,016 |
Fin instanciation : | 125 | 0 |

Le centre et les plates-formes utilisent les mouvements du pointeur graphique pour discuter des chemins possibles pour le vaisseau non identifié. On montre seulement quelques mises à jour du pointeur. En pratique, il y aurait normalement un grand nombre de points transmis pour retracer le mouvement des pointeurs sur les écrans.

Depuis le centre :

Suivi du pointeur : | 137 | 16 | 0 | 0,39 0,032 |
 Suivi du pointeur : | 137 | 16 | 0 | 0,388 0,035 |
 Suivi du pointeur : | 137 | 16 | 0 | 0,388 0,039 |
 Suivi du pointeur : | 137 | 16 | 0 | 0,386 0,04 |

Depuis une des plates-formes :

Suivi du pointeur : | 137 | 16 | 0 | 0,22 0,016 |
 Suivi du pointeur : | 137 | 16 | 0 | 0,222 0,159 |
 Suivi du pointeur : | 137 | 16 | 0 | 0,233 0,157 |
 Suivi du pointeur : | 137 | 16 | 0 | 0,24 0,155 |

Le centre dessine maintenant le chemin prévu que va suivre le vaisseau non identifié. Cette fois, la trace du pointeur est enregistrée sur l'écran en traçant une ligne.

Suivi du pointeur : | 137 | 16 | 1 | 0,388 0,038 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,386 0,038 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,386 0,052 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,375 0,078 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,369 0,105 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,361 0,125 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,352 0,144 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,351 0,156 |
 Suivi du pointeur : | 137 | 16 | 1 | 0,35 0,16 |

Une plate-forme déplace les icônes des navires US sur les positions d'interception.

Transformation d'objet : | 126 | 55 | 15 | US_CONSTITUTION | 1,0 0,0 0,16 0,0 1,0 -0,046 |
 Transformation d'objet : | 126 | 53 | 13 | US_NEW_JERSEY | 1,0 0,0 0,113 0,0 1,0 -0,034 |

Le centre fait un agrandissement sur la zone d'interception afin d'obtenir une vue plus précise pour la suite de la discussion.

Fenêtre de la station de travail : | 71 | 24 | 0,286 0,403 0,077 0,177 |

Les deux plates-formes indiquent leurs portées de frappe en utilisant des bandes de surbrillance circulaires centrées sur chaque navire. Pour chaque plate-forme, on montre d'abord le point de référence d'écho puis les deux points de retour d'écho. Normalement, il y aurait un plus grand nombre de points de rétroaction.

Bande de surbrillance : | 138 | 10 | 0 | 0,335 0,125 |
 Bande de surbrillance : | 138 | 10 | 3 | 0,35 0,128 |
 Bande de surbrillance : | 138 | 10 | 3 | 0,37 0,128 |
 Bande de surbrillance : | 138 | 10 | 0 | 0,384 0,13 |
 Bande de surbrillance : | 138 | 10 | 3 | 0,367 0,128 |
 Bande de surbrillance : | 138 | 10 | 3 | 0,346 0,129 |

Une fois l'accord réalisé sur la stratégie d'interception, le centre revient à l'image d'origine, plus vaste.

Fenêtre de la station de travail : | 71 | 24 | 0,0 0,5 0,0 0,375 |

Le centre termine la conférence :

Fin d'élément : | 0 | 0 |

À la fin de la conférence, les images finales restent visibles sur les écrans. De plus, les éléments PIGCF seront enregistrés dans leur totalité afin de revoir si nécessaire la session de conférence. L'enregistrement de la conférence pourrait aussi être envoyé à d'autres localisations au titre d'un message multimédia.

Références

[1] J. D. Day and H. Zimmermann, "The OSI Reference Model", Proceedings of the IEEE, V 71, N 12 ; décembre 1983,

pp 1334-1340.

- [2] W. Pferd, L. A. Peralta and F. X. Prendergast, "Interactive Graphics Teleconferencing", IEEE Computer, V 12, N 11 ; novembre 1979, pp 62-72.
- [3] K. S. Sarin, "Interactive On-Line Conferences", Ph.D. Diss. MIT, Dept. of EE and CS, 1984.
- [4] S. Randall, "The Shared Graphic Workspace: Interactive Data Sharing in a Teleconference Environment", Proceedings CompCon 82 Fall, IEEE Computer Society, pp 535-542.
- [5] G. Heffron, "Teleconferencing Comes of Age", IEEE Spectrum, Oct. 1984, pp 61-66, pp 61-66.
- [6] R. W. Hough and R. R. Panko, "Teleconferencing Systems: A State-of-the-Art Survey and Preliminary Analysis", SRI International, Menlo Park California, SRI project 3735, avril 1977.
- [7] C. W. Kelly III, "An Enhanced Presence Video Teleconferencing System" Proc. CompCon, 20-23 septembre 1982, Washington D.C., pp 544-551.
- [8] J. Vanglian, "Communication privée", Commentaires sur l'acceptabilité du vidéotex pour la communication graphique en ligne.
- [9] ANSI Technical Committee X3H, "Draft Proposal: Virtual Device Metafile", X3.122, X3 Secretariat, CBEMA, Washington, D.C.
- [10] American National Standards Committee X3H3, "Virtual Device Interface", X3 - Information Processing Systems, Document de travail du 2 janvier 1985, disponible auprès de Computer and Business Equipment Manufacturers Association, Washington D.C.
- [11] E. Van Deusen, "Graphics Standards Handbook", CC Exchange 1984, P.O. Box 1251, Laguna Beach, CA 92652.
- [12] J. D. Foley and A. Van Dam, "Fundamentals of Interactive Computer Graphics", Addison-Wesley, 1982.
- [13] American National Standards Committee X3H3, "GKS -- 3D Extensions", X3 - Information Processing Systems, Document de travail du 16 novembre 1984, disponible auprès de Computer and Business Equipment Manufacturers Association, Washington D.C.
- [14] ANSI Technical Committee X3H3, "Draft Proposal: Graphical Kernel System", X3.124, X3 Secretariat, CBEMA, Washington, D.C.
- [15] G. Enderle, K. Kansy, and G. Pfaff, "Computer Graphics Programming", Springer-Verlag, 1984.
- [16] ISO/DIS 6093.2, "Traitement de l'information - Représentation des valeurs numériques dans les chaîne de caractères pour les échanges d'information", ISO/TC 97, 1984-01-19 ; disponible aussi auprès de AFNOR, Saint Denis. France