



HAL
open science

ITAMEX, Extension de X pour le transfert et la visualisation d'images

Patrice Lutmann, Richard Castanet, Bernard Cousin

► **To cite this version:**

Patrice Lutmann, Richard Castanet, Bernard Cousin. ITAMEX, Extension de X pour le transfert et la visualisation d'images. Convention Unix 92, systèmes ouverts : l'interopérabilité, Sylvain Langlois, Mar 1992, Paris, France. pp.131 - 140. hal-01490658

HAL Id: hal-01490658

<https://hal.science/hal-01490658>

Submitted on 15 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ITAMEX

Image Transfert Access and Management Extension to X

Extension de X pour le Transfert et la Visualisation d'Images

Lutmann Patrice,
Castanet Richard,
Cousin Bernard,
LaBRI (Laboratoire Bordelais de Recherche en Informatique)*,
351 cours de la Libération, 33405 Talence, FRANCE

E-mail: lutmann@labri.greco-prog.fr, lutmann@geocub.greco-prog.fr

RESUME

Du fait de l'impact croissant de l'Image dans l'informatique et de l'explosion des réseaux de communication, les applications informatiques nécessitent de plus en plus le transfert et la visualisation d'images. Pour réaliser ces applications, il faut posséder un service de transfert et de visualisation d'images numériques.

Le but de cet article est de décrire un nouveau service appartenant à la couche application du modèle de référence OSI: ITAMEX. Ce service réunit ITAM (transport des données) et EX (extension du système X). Il s'intègre parfaitement dans le monde des stations de travail graphiques sous UNIX, mais aussi sur de plus gros systèmes, le tout faisant partie de systèmes répartis interconnectés par des réseaux de communication. L'article présente les différentes possibilités d'intégration d'ITAMEX dans l'environnement X (X Window System) et évoque les divers problèmes qui sont ainsi soulevés.

ABSTRACT

Because of the growing impact of image in computer field and of the increasing use of communication network, computer applications need more and more often to transfer pictures and visualise them. In order to develop those applications, you need a pictures transfer service and a pictures visualisation service.

The aim of this article is to describe a new service which is part of the OSI model (application layer). The service ITAMEX joins together ITAM (image data transfer) and EX (Extension to X). It is entirely integrated into the UNIX graphical workstations world. The whole is part of a distributed environment which consists in a set of computers interconnected with a communication network. This article describes the different functionalities of the ITAMEX integration in the X Window System and shows the diverse issues which are arised.

(*) Unité de recherche 1304 associée au CNRS

I - INTRODUCTION

1 . Etats de faits

L'évolution de la communication intègre de plus en plus les images, ceci résultant simplement du fait qu'un dessin vaut mieux qu'un long discours et que l'évolution de la technique actuelle met l'image à la portée de tous.

2 . Les besoins actuels

a) Le transfert d'images

Dans le monde informatique en particulier, on constate que dans les systèmes répartis de plus en plus d'applications nécessitent un transfert d'images numériques [Lorig 83]. La réalisation de telles applications pose de nouveaux problèmes dus à la nature même de l'information à véhiculer. Il faut alors, pour réaliser ces applications, posséder un service de transfert d'images numériques adéquat. Le service ITAM (Image Transfer Access and Management) [Navarro 91] répond à un tel besoin en définissant un nouveau service appartenant à la couche application du modèle de référence OSI (Open Systems Interconnection) [ISO 7498].

b) La visualisation d'images

Les applications utilisant un transfert d'images sont principalement des sauvegardes ou des visualisations d'images [Lebon 85]. Seulement, pour visualiser une image numérique sur un périphérique graphique (généralement au moins une fois un écran), il est nécessaire d'adapter cette image à ce périphérique. Le service ITAMEX (ITAM Extension to X) propose un ensemble de services permettant de transférer et de visualiser des images. Ce service fournit une interface ad hoc de visualisation d'images dans l'environnement X (X Window System) [MIT 85].

II - ITAM

1 . Les problèmes du transfert d'images

a) Le débit

Lorsqu'on parle de transfert d'images, il s'agit la plupart du temps de véhiculer un gros volume d'information en un laps de temps assez court. Pour cela, il faut un débit utile conséquent pour avoir des temps de réponse acceptables. Pour résoudre ce problème, deux options sont possibles. On peut utiliser des réseaux à très haut débit qui conviennent au transfert d'images, comme par exemple FDDI [ANSI X3.139] qui propose des débits de 100 Mbits/s. Dans le cas où l'on ne dispose pas de débit suffisant (citons RNIS [Dicenet 88], avec 2x64 Kbits/s), on peut palier à cette faiblesse en utilisant des techniques de compression de données adaptées à l'image, comme par exemple le standard JPEG [ISO 10918]. On peut aussi combiner ces deux méthodes.

b) La représentation des images

La représentation d'une image numérique dépend de l'origine de cette image. En effet, le codage d'une image numérique est différent selon que l'image provienne d'un dispositif numérique de saisie (par exemple un scanner) ou qu'elle soit le résultat d'une application graphique (par exemple une image de synthèse). Pourtant, dans chacun des cas, l'image a une sémantique propre qui donne son sens à l'image. De plus, entre la genèse de l'image avec son codage propre et l'image stockée, il y a souvent une étape intermédiaire qui adapte l'image brute au support de stockage, ne serait-ce que pour rendre l'image exploitable ultérieurement. Il y a ainsi une multitude de façon de représenter une image et l'on ne compte plus les standards de stockage d'images (TIFF [Aldus 88], GIF [CompuServe 87], FTICRP [Hofmann 87], etc). Par abus de langage pour mieux comprendre l'exposé de cet article, on assimilera l'image au fichier de données représentant cette image.

c) L'hétérogénéité des matériels

L'hétérogénéité des matériels informatiques graphiques pose le problème de la restitution des images [Blum 90]. En effet, une image n'a pas de sens si elle n'est que stockée. On a besoin de pouvoir restituer l'image sur un support afin de la visualiser. La difficulté majeure que l'on rencontre à ce

niveau est d'adapter l'image au périphérique de visualisation (par exemple afficher une image couleur sur un écran monochrome). Il n'y a pas à l'heure actuelle de périphériques de visualisation standardisés permettant de restituer sans transformation une image sur une large gamme de matériels. Cette difficulté est encore accrue si l'on tient compte de la diversité des formats de stockage d'images numériques.

d) Le mode de transfert

Le transfert d'image est plus complexe qu'un simple transfert passif d'un fichier d'images entre deux sites. Si la nature de l'information transférée est bien définie, le mode de relation peut être bipoint ou multipoint et le mode de service peut être connecté ou non-connecté. En effet, si le transfert le plus utilisé est le transfert point à point en mode connecté, il ne suffit plus pour des applications multimédia [Hughes 90]. De plus, ce type d'application requiert bien souvent un transfert progressif ou sélectif pour mieux cibler l'information utile.

e) Bilan des problèmes

Pour être efficace, le transfert d'images doit tenir compte non seulement des caractéristiques de l'image (la sémantique) mais aussi des ressources disponibles sur le site de stockage des images et sur le site demandeur d'images. Il faut alors déterminer la représentation de transfert des images en fonction des capacités de transformation de chaque site pour préserver la sémantique de l'image. La représentation de transfert doit aussi minimiser le volume d'information à transmettre afin de ne pas surcharger le réseau.

2 . Le service de transfert ITAM

a) L'objectif d'ITAM

Le but du service ITAM est de fournir à l'utilisateur un transfert d'images négocié qui respecte la sémantique des images tout en minimisant le volume à transmettre. En tenant compte des possibilités des sites acteurs du transfert, le service ITAM fournit un transfert qui minimise le volume des informations à transférer en ne transmettant que des informations utiles. Pour réaliser cela, le service ITAM propose un ensemble de fonctionnalités comme la négociation de la représentation de transfert, la négociation de la qualité de transfert d'images, un suivi interactif du transfert en cours... Ces fonctionnalités sont impossibles à réaliser avec le seul service de transfert de fichier FTAM (File Transfer Access and Management) [ISO 8571].

b) Le mode de transfert d'ITAM

Le service ITAM se limite au mode connecté bipoint et permet de réaliser des transferts progressifs et sélectifs.

3 . Les composantes d'ITAM

Le service ITAM est un service d'application qui fait appel aux éléments de service ACSE (Association Control Service Element) [ISO 8649, 8650], VT (Virtual Terminal) [ISO 9040, 9041] et FTAM.

a) Le service ACSE

Tout échange de données au niveau application se réalise au travers d'une association. Pour gérer cette association, l'élément de service de contrôle d'association est présent dans tous les objets d'association simples et est utilisé avec d'autres éléments de service d'application. La norme de définition de service [ISO 9649] et la norme de spécification du protocole [ISO 9650] ACSE décrivent une association d'application comme une relation entre deux entités d'application pour l'échange d'information de contrôle de protocole d'application. ITAM utilise le mode de fonctionnement normal du service ACSE pour supporter les applications en mode connecté.

b) Le service VT

Le service VT propose des primitives pour la définition et la négociation des paramètres de profil. Un profil décrit la représentation d'un terminal virtuel. Dans le cadre du transfert d'image, on s'intéressera surtout aux terminaux ayant des capacités graphiques. Ainsi, grâce à la définition de profils

paramétrés, une application peut déterminer les paramètres optimum pour le transfert des images. Ces paramètres sont déterminés pendant la phase de négociation du service ITAM.

c) Le service FTAM

Le service FTAM permet le transfert, l'accès et la gestion de fichiers. Il est utilisé pour accéder et transférer l'ensemble des données préalablement sélectionné par le service ITAM.

4 . Mode de fonctionnement

a) Deux associations d'application

Pour pouvoir profiter d'un service de transfert souple et interactif, il est nécessaire de séparer les commandes de gestion du transfert de données des données à transférer. Pour cela, le service ITAM utilise deux associations d'application: l'association de commandes et l'association de données. L'association de commande est utilisée pour le négoce du transfert (phase d'établissement de connexion, sélection, ouverture) et pour la gestion du transfert en cas de transfert interactif. L'association de données est utilisées pour le transfert effectifs des données (des images numériques).

b) Les caractéristiques du protocole

Le transfert d'image du service ITAM comporte une étape de préparation (ou de négoce) suivie d'une étape de transfert. L'étape de préparation détermine le site accepteur, la ressource distante et la représentation de transfert. L'étape de préparation permet ainsi de déterminer le profil optimum pour l'échange des données. Ensuite, le transfert des données est assuré par l'élément de service d'application FTAM.

III - ITAMEX

1 . ITAM Extension to X

a) Réalisme d'ITAM

L'objectif avoué d'ITAM est de pouvoir être porté sur la majorité des stations de travail, généralement UNIX, chacune pouvant être reliée à divers écrans graphiques. Là, l'environnement graphique le plus utilisé et le mieux accepté est le standard X.(X Window System du Consortium MIT). Là aussi, malheureusement, FTAM est pratiquement inexistant. Ces deux constats nous amènent à reconsidérer ITAM et à proposer ITAMEX. Du point de vue de l'utilisateur, ITAMEX proposera les mêmes services qu'ITAM. Les différences fondamentales sont l'abandon de FTAM pour le transfert des images au profit d'un service plus abordable (par exemple FTP) et un service supplémentaire de visualisation d'images intégré à X. Les développements ultérieur d'ITAMEX porteront sur la diffusion d'images, le mode multipoint et le mode non connecté.

b) L'apport d'ITAMEX

ITAMEX offre à l'utilisateur un ensemble de services et de fonctionnalités lui permettant de transférer, visualiser et manipuler des images, et ce, à faible coût sur une très large gamme de matériel (stations de travail graphiques UNIX). ITAMEX contribue ainsi à une meilleure interopérabilité entre les systèmes informatiques, l'image et l'homme, consommateur d'images. De plus, en offrant une interface conviviale et transparente, il débarrasse l'utilisateur des contraintes liées au transfert d'image et des problèmes d'adaptation de l'image au périphérique de sortie (l'écran X).

c) Pourquoi X ?

Actuellement, l'environnement graphique le plus répandu est le système de multifenêtrage graphique X Window System, plus connu sous le nom de X. Ce standard de l'industrie offre la possibilité au programmeur de développer des interfaces utilisateurs graphiques portables. L'un des grands attraits de X réside dans son architecture unique indépendante du matériel. X permet à une application d'afficher des fenêtres contenant textes et graphiques sur n'importe quel type de matériel supportant le protocole X, et ce, sans avoir à recompiler l'application. Cette indépendance vis-à-vis du matériel, doublée de la reconnaissance de X comme un standard, offre la possibilité aux applications

basées sur X de pouvoir fonctionner efficacement dans un environnement hétérogène regroupant de nombreux matériels informatiques, du PC au CRAY.

2. X Window System

a) Modèle

L'architecture de X est basée sur le modèle client-serveur travaillant en parfaite coopération. Le client est un demandeur de service que le serveur se charge de réaliser.

Un processus unique, le serveur, contrôle les périphériques d'entrée-sortie (en règle général, un écran, un clavier et une souris). C'est le serveur qui crée et manipule les fenêtres sur l'écran, dessine le texte et les graphismes, gère le clavier et la souris. Le serveur fournit ainsi une interface portable entre toutes les applications et le matériel.

Une application utilisant les facilités offertes par le serveur X est appelée client. Un client communique avec le serveur X via une connexion réseau utilisant un protocole de niveau transport orienté connexion. X permet d'utiliser plusieurs protocoles réseaux dont TCP/IP, DECnet et Chaos. Plusieurs clients peuvent se connecter à un même serveur, et un même client peut se connecter à plusieurs serveurs.

En plus de la transparence matérielle, l'architecture répartie du système X permet au serveur et aux clients de s'exécuter sur des machines distinctes localisées n'importe où sur un réseau.

Fondamentalement, le serveur X peut être considéré comme une machine abstraite prestataire de services, même si le serveur en lui-même est fonction du type de matériel sur lequel il est implémenté. L'interface avec les clients, le protocole X, reste toujours le même.

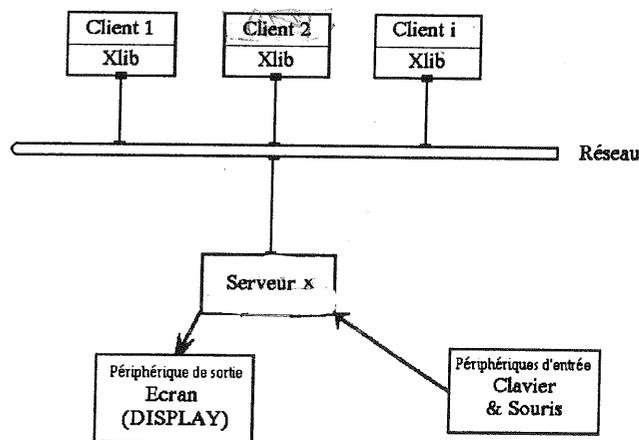


Fig 1 - Le modèle Client/Serveur de X

b) Les composantes du protocole X

Côté client, l'encodeur, du protocole X est la Xlib: c'est l'interface responsable de la mise en forme des requêtes du client en paquets propices à la transmission par le réseau et de la communication avec le serveur.

Le serveur, de son côté, reçoit les paquets envoyés par l'interface Xlib d'un client et les traite pour exécuter les ordres demandés par le client.

Le protocole X définit quatre types de paquets:

(1) requête erveur --> client)	(client --> serveur)	(2) réponse	(s
(3) erreur erveur --> client)	(serveur --> client)	(4) événement	(s

Le client émet des paquets de type (1) et reçoit des paquets de type (2), (3) et (4). Les paquets (2) et (3) sont en général des réponses à des requêtes (1), tandis que les paquets de type (4) sont envoyés par le serveur de manière asynchrone pour signifier au client les modifications intervenues dans l'environnement (événement).

c) Restriction fondamentale

La politique décisionnelle du serveur repose sur le principe de base suivant: "le serveur ne fait que servir le client et en aucun cas ne prend de décision quant aux requêtes des clients". Il ne fait que gérer les ressources graphiques attribuées à chaque client. C'est à chaque application (client du serveur X) de gérer son environnement et de veiller à ne pas monopoliser les ressources du serveur à elle seule [ICCCM].

d) Gestion de multiples écrans

Le serveur X est capable de gérer plusieurs écran de capacités différentes (résolution graphique, nombre de couleurs, etc). Ceci est rendu possible par une structure d'encapsulation des différents périphériques physiques fournissant une interface abstraite aux divers matériels graphiques sous-jacents.

e) Transparence réseau

Dans le but d'offrir un accès homogène à travers le réseau entre toute les machines, chacune ayant son architecture propre, le protocole X rejette à la fois les types flottants et les formats de couleur, informations difficilement portables dans un système réparti. Lors de l'envoi de paquets à travers le réseau, le système devra maintenir la cohérence de l'information contenue dans les paquets transmis.

f) Philosophie du concept X

La philosophie de base de X est de fournir des services de haut niveau permettant de faire du graphisme 2D élémentaire et de manipuler de manière efficace des fenêtres graphiques.

L'environnement de travail supposé nécessaire par X consiste en un ensemble de stations de travail graphiques suffisamment performantes, reliées entre elles par un réseau à débit élevé (quelques Mbits/s, par exemple ethernet), de manière à pouvoir partager les ressources graphiques et la puissance de calcul efficacement.

3 . Modèles de service avec ITAMEX

a) L'impact de X sur le service ITAMEX

Le fait d'apporter un service de visualisation ad hoc d'images prolongeant le service de transfert impose une association supplémentaire: c'est l'association de visualisation. Cette association sera responsable du transfert de données (l'image) entre l'application et le serveur X. Elle définira aussi les paramètres de visualisation afin de restituer une image la plus fidèle possible par rapport à l'image originale. De fait, l'application devient un client du serveur X.

Note:

Le service ITAM négocie le transfert de l'image de manière à trouver la meilleure adéquation possible entre l'image sur le site de stockage et les capacités du site demandeur. Comme nous venons de le voir, dans le monde X, une même application peut être client de plusieurs serveurs, chacun d'eux étant associé à un écran, et chacun de ces écrans possède ses propres caractéristiques (résolution graphique, nombre de couleurs disponibles, etc). Ainsi, la négociation de transfert dans le but de trouver le meilleur profil peut être rendue caduque dans le cas où l'application décide de visualiser l'image sur des écrans ne possédant pas les mêmes caractéristiques.

b) Les deux principaux modèles

L'intégration d'ITAMEX dans le monde X peut se réaliser extra-X ou intra-X.

Dans le premier cas, l'implémentation d'ITAMEX se résume à implémenter ITAM pour le transfert des images, avec en plus un ensemble de fonctionnalités permettant de se connecter à un serveur X et d'afficher des images. Cela nécessite alors deux relations bipoints, l'une entre l'application et le site de stockage (pour les associations de commande et de données), l'autre entre l'application et le serveur X (pour l'association de visualisation). Voir la figure 2. Il y a un double flux de données entre le site de stockage et l'application, et entre l'application et le serveur. Dans ce cas, le serveur X n'a qu'un rôle passif qui consiste à afficher des images dans des fenêtres.

Dans le second cas, l'implémentation d'ITAMEX est répartie entre les divers acteurs. Le serveur X joue alors un rôle actif. Le service ITAM n'est plus localisé à l'application, mais il est partagé entre l'application (qui est un client, rappelons-le) et le serveur X. Ce mode de fonctionnement nécessite de greffer une partie du service ITAMEX sur le serveur (c'est le principe des extensions). Cela permet de n'avoir qu'un flux de données entre le site de stockage et le site de visualisation, à savoir le serveur X. L'application a ainsi un rôle essentiellement de commandes, l'une ayant trait au transfert d'images, l'autre conditionnant la visualisation. Ce procédé requiert trois relations bipoints. La première concerne l'association de visualisation entre l'application et le serveur X. La deuxième concerne l'association de commande du transfert d'images entre l'application et le site de stockage. La troisième concerne l'association de données entre le serveur X et le site de stockage. Voir la figure 3.

c) Modèle unilatéral client (bi-connexion)

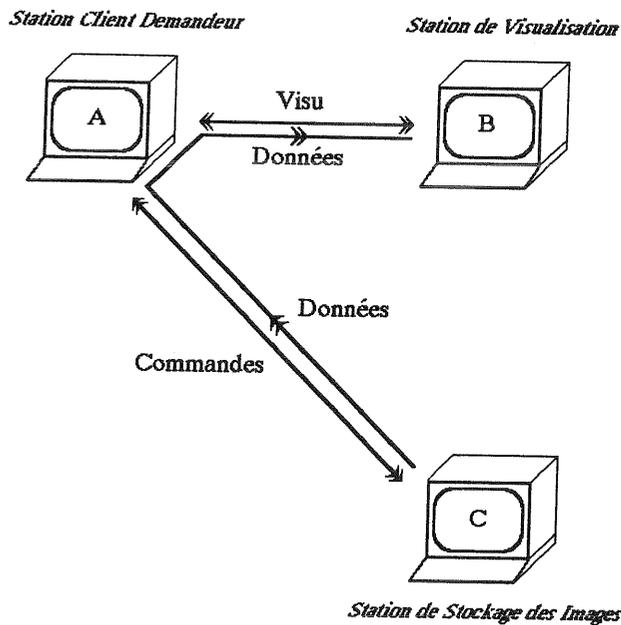


Fig 2 - Modèle bi-connecté

L'application se trouvant sur le site demandeur effectue le négoce et le transfert d'images avec le site de stockage au travers de deux associations d'applications: l'association de commande et l'association de données. Elle demande au serveur la visualisation de l'image ainsi transférée au travers d'une troisième association de visualisation. Les données de l'image sont alors envoyées au serveur X pour un affichage sur l'écran.

- Avantages

L'application dispose directement des données et accède donc directement à l'image. Cela permet aussi une implémentation d'ITAM simplifiée si on se repose sur le service d'application FTAM pour le transfert des données. L'application a de plus le contrôle total sur le transfert des données.

- Inconvénients

L'application est seule responsable des trois associations d'application et est totalement monopolisée pour la commande et le transfert des données. Comme nous l'avons vu précédemment, l'image récupérée sur le site demandeur doit être dupliquée sur le site de visualisation: il y a donc deux fois le volume à transférer entre le site de stockage et le site demandeur, et entre le site demandeur et le site de visualisation. L'affichage n'est pas optimum car l'application n'est pas maître du périphérique d'affichage du site de visualisation. De plus, ce mode de service est inadapté et inefficace pour la diffusion d'images (Cf Note III.3.a).

d) Modèle bilatéral client-serveur (tri-connexion)

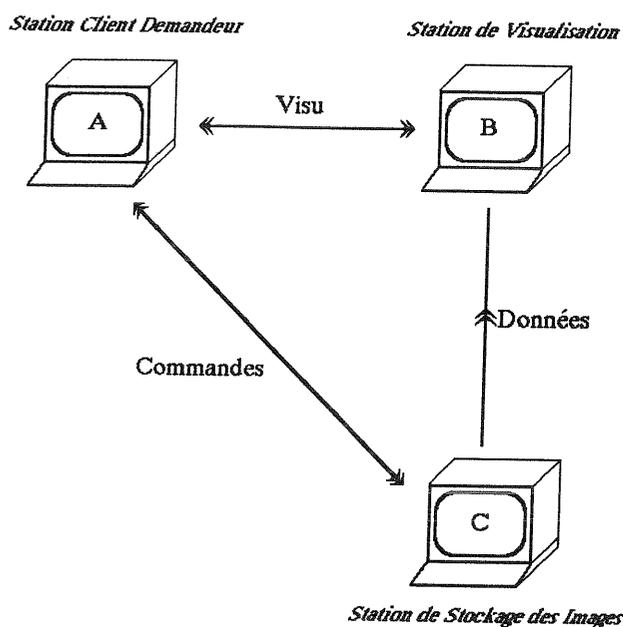


Fig 3 - Modèle tri-connecté

L'application effectue le négoce du transfert de fichier avec le site de stockage au travers d'une association d'application: l'association de commande. Le transfert de données proprement dit s'effectue directement entre le site de visualisation et le site de stockage au travers d'une association de données non nécessairement de niveau application. La visualisation s'effectue au travers d'une association de visualisation comprenant à la fois la commande de visualisation et un service de réception facultatif des données du site de visualisation vers l'application. Le serveur X étendu prend directement en charge l'affichage de l'image à l'écran selon les besoins de l'application.

- Avantages

L'implémentation d'ITAM peut être simplifiée pour le transfert des données en n'utilisant pas un service d'application tel que FTAM mais un service moindre comme FTP. Le site de visualisation étant aussi le site de réception des données, il permet une visualisation immédiate de l'image, sans intermédiaire et adaptée au périphérique. Il n'y a pas non plus de duplication inutile de données entre l'application et le site de visualisation si l'application n'a pas besoin de disposer de l'image en local. De plus, L'application est libérée de la gestion bout en bout du transfert d'images.

- Inconvénients

L'application n'a pas le contrôle effectif du transfert des données et ne dispose pas immédiatement de l'image. Elle n'est pas non plus maître de la visualisation de l'image (traitement limité par les services du site de visualisation). L'application n'a pas le contrôle du transfert ce qui limite l'interactivité du transfert, à moins d'étendre l'association de visualisation.

e) Implications

La réalisation du modèle décrit en c) implique, dans le cadre de travail avec X, le développement d'une interface application avec, au niveau du site demandeur et du site de stockage, une interface de présentation des données requise par la négociation du transfert. On peut néanmoins se baser sur des services existant pour le transfert des fichiers images, soit en utilisant FTAM, soit plus simplement FTP.

Le modèle décrit en d) nécessite le développement d'une interface application pour le site demandeur et une extension du système X pour le site de visualisation, d'une part pour l'affichage, d'autre part pour le transfert des données. L'application n'étant plus responsable du transfert des données, on peut utiliser les services des couches basses plus performantes pour le transfert.

4 . Choix du modèle

Le choix du modèle est conditionné par un soucis d'efficacité, d'extensibilité, de viabilité et de fiabilité. Il faudra aussi que le système retenu soit en mesure d'afficher avec un délai acceptable des séquences animées et pouvoir supporter la diffusion d'images.

a) Extension du serveur

Le modèle d) est le mieux adapté au service ITAMEX car il fournit une interface de visualisation, implicitement imposée par ITAMEX, qui soit adaptée au périphérique de visualisation et la plus performante possible. Il repose sur le protocole X (futur standard ISO ?) bien adapté aux réseaux hétérogènes.

L'extension du serveur est rendue nécessaire si l'on tient compte du facteur d'efficacité et de la redondance d'information inutile (double flux de données). C'est le serveur qui prend en charge et le transfert des données avec son vis-à-vis (un serveur X ou un serveur aveugle) et l'affichage de l'image. Etant directement lié au périphérique physique, il est à même d'afficher le plus efficacement possible l'image. Cela est impératif dans le cadre des séquences animées. De ce fait, l'application est libre pendant le transfert et l'affichage des images et peut ainsi utiliser le temps machine qui aurait été monopolisé.

De plus, au cas où l'image n'est pas requise en local par l'application, il n'y a pas de duplication de l'information (souvent de gros volumes, de l'ordre du méga-octets et plus), et donc pas de monopolisation du support de transfert au cas où l'application (client) et le serveur ne sont pas localisés sur le même site.

b) Modèle client-serveur de l'extension

L'extension du système X comprend à la fois une extension du serveur (extension des services) et une extension de l'interface application qu'est la Xlib. Voir la figure 4.

L'extension du serveur comprend deux parties, l'une concerne l'association de données, l'autre concerne la visualisation des images. Ces deux extensions sont complétées par un mécanisme d'événement permettant un dialogue asynchrone avec l'application.

Le côté client est symbolisé par une extension de la Xlib, l'extension consistant principalement à implémenter la phase d'initialisation et de négociation, de récupération des données en provenance du serveur, et à la mise en place d'un "handler" d'actions sur événement.

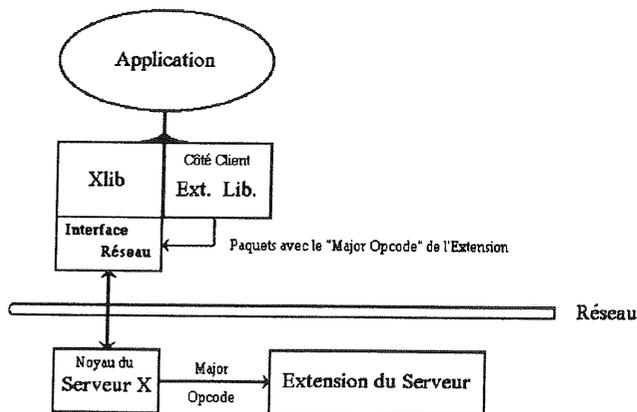


Fig 4 - Principe d'une extension X

IV - Conclusion

1 . Implémentation d'ITAMEX

a) Niveau de X requis

On utilise la version 11.4 du système X (X11 R4) avec l'extension permettant d'utiliser les segments de mémoire partagée.

b) Architecture de tests

Le modèle d) a été retenu pour une implémentation d'ITAMEX. L'implémentation actuelle correspond à un modèle hybride mieux adaptée aux tests et à la validation. L'extension du serveur est alors détachée de celui-ci et devient un client privilégié qui s'exécute sur la même CPU que le serveur. C'est alors un processus indépendant. On utilise les segments de mémoire partagée comme moyen d'échange de données rapide entre le serveur et l'extension-client. Cette extension a à sa charge le transfert d'images avec le site de stockage et la conversion de ces images en pixmapes dédiées à X (les XImages). Elle est commandée par l'application pour la gestion du transfert des données et pour l'affichage. Sur le site de stockage, on implémente un serveur de données aveugle disposant d'une archive d'images et d'un annuaire technique résumant les caractéristiques de celles-ci. L'application sélectionne l'image à transférer, commande l'extension pour le transfert et décide de l'affichage.

c) Les événements

Nous l'avons vu précédemment, l'application n'est ni maître du transfert ni maître de l'affichage. Pour remédier à cela, on utilise un mécanisme d'événements asynchrones permettant à l'application et à l'extension de dialoguer. Cela est indispensable pour synchroniser l'application et l'extension, mais aussi pour avertir l'application en cas de problèmes, de fin de transfert, de manque de ressources, etc...

2. Cadre d'essais

Pour comparer les deux modèles au niveau des performances, le modèle c) et le modèle hybride ont été implémentés. Le cadre de test est constitué d'un ensemble de trois stations de travail graphiques d'une puissance suffisante, sous UNIX avec X11 R4. Elles sont toutes trois reliées par un réseau ethernet. La première machine joue le rôle du site demandeur, la seconde le site de visualisation et la dernière le site de stockage.

Le test porte sur une séquence d'images animée (50 images) d'une résolution 256x256x8 soit 524288 bits (une image de taille 256 par 256 pixels, le pixel étant ici codé sur 8 bits). Le codage de ces images est un codage par plage de valeurs égales et font en moyenne 20 Ko. Le débit efficace local est de 20 images/s.

Les résultats sont les suivants: dans le cas du modèle c), on arrive à un débit efficace de 8 images/s tandis qu'avec le modèle hybride on atteint un débit efficace de 18 images/s. En doublant la résolution (512x512x8), les débits tombent à moins de 2 images/s et moins de 4 images/s respectivement.

3. Conclusion

ITAMEX répond aux besoins croissants de transfert et de visualisation d'images, en offrant à l'utilisateur un ensemble de services et de fonctionnalités de haut niveau lui permettant de transférer et de visualiser des images numériques de manière transparente. Cela est rendu possible grâce au standard X (pour la partie graphique) et à ITAM (pour la partie transfert des données). ITAMEX met ainsi l'image à la portée de tous, utilisable sur une très large gamme de matériels informatiques.

IV - Références et Bibliographie

- * [ICCCM] InterClient Communications Conventions Manual, Documentation X11 MIT
- * [Lorig 83] B. Lorig, "Les services d'images", La Recherche, No 144, Mai 1993
- * [Lebon 85] M. Lebon, "TROPICS: Transfert d'images fixes avec textes, via satellite (projet Nadir)", Bulletin de liaison de la recherche en informatique et automatique, No 103, 1985
- * [Dicenet 88] G. Dicenet, "Le RNIS, techniques et atouts", Editions Masson, 1988
- * [MIT 85] X Window System, Trademark of the Massachusetts Institute of Technology (USA).
- * [Hughes 90] L. Hughes, "A Hypermedia System With Educational Applications: The Edmund Hypermedia Research Project", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990
- * [Blum 90] C. Blum, "Fast Image Communication and Presentation in a Distributed LAN Environnement", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990

- * [Hofmann 87] G. R. Hofmann, "FTCRP - Format for the Transfert of Colored Raster Picture", Version 0, GI 4.1.4, Fachgespräch, Berlin 1987
- * [CompuServe 87] GIF, "Graphics Interchange Format", (c) CompuServe 1987,1989
- * [Aldus 88] TIFF, "Tag Image File Format", Spec. Revision 5.0, (c) Aldus & Microsoft Corp, Seattle 1988
- * [Navarro 91] X. Navarro, "Un service d'application de transfert d'images", Thèse D.I, Université de Bordeaux, Mars 1991

- * ANSI X3.139 Fiber Distributed Data Interface (FDDI)
- * ISO 2093, 2094 Transaction Processing (TP)
- * ISO 7498 Open Systems Interconnection (OSI)
- * ISO 8571 File Transfert Access and Management (FTAM)
- * ISO 8649, 8650 Association Control Service Element (ACSE)
- * ISO 8831, 8832 Job, Transfert and Management (JTM)
- * ISO 9040, 9041 Virtual Terminal (VT)
- * ISO 9072 Remote Operation Service (ROS)
- * ISO 9804 Concurrency, Commitment and Recovery (CCR)
- * ISO 10918 (CD) Joint Photographics Expert Group (JPEG), Rev. 9.6, Janvier 1991

- * R. Castanet et O. Rafiq, "Ingénierie des protocoles", CFIP 88, Bordeaux, Editions Eyrolles, Septembre 1988
- * Hsien Ching Kelvin Sung, "UIPEX: a 3D Graphics extension to the X Window System", Thesis 1990, University of Wisconsin, University of Illinois
- * R.W. Scheffler, J. Gettys, "The X Window System", ACM Trans. on Graphics, Vol 5, No 2, Avril 1986, pp 79-109
- * G. Rogers, K. Sung, W. Kubitz, "Combining Graphics and Windowing Standard", Graphics Workstation Research Group, Tech. Ref. No 1, Janvier 1989, University of Illinois
- * "X11", "Protocole X", Documentation X11 MIT
- * B. McCormick, T. DeFanti, M. Brown, "Visualization in Scientific Computing", SIGGRAPH Computer Graphics Newsletter Vol 21, No 5, Octobre 1987
- * A.S. Tanenbaum, R. van Renesse, "Distributed Operating System", ACM Computing Surveys, Vol 17, No 4, Décembre 1985
- * Lee Tibbert, R. Daniel Bergeron, "Graphics Programming for Knowledge-guided Interaction", Eurographics 85
- * M.I. Garrett, J.D. Foley, "Graphics Programming using a Database System with Dependency Declaration", ACM Trans. on Graphics, Vol 1, No 2, Avril 1982, pp 109-128
- * D.B. Neal, "Tutorial on Graphics and Database", SIGGRAPH'83 Tutorial Notes 20, Juillet 1983
- * F. BALL, "A Multimedia Network Interface", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990
- * T. MIYAMOTO, "Multimedia Server Architecture for Broadband Networks", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990
- * G.S. Blair, "Engineering Support for Multimedia Application in Open Distributed Processing", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990
- * S.R. AHUJA, "Multipoint Protocol Aspects of Audiovisual Teleconferencing via 2x64 Kbits/s Chanel", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990
- * C. Bertin, Y. Sureur, "Protocols for Multimedia Interactive Applications", 3rd IEEE Com. Int. Workshop on Multimedia Communications (Multimedia'90), Bordeaux, Novembre 1990
- * N. Dal Degan, P. Migliorati, S. Pozzi, V. Trecordi, "Still Images Retrieval from a Remote Database", 1st Int. Conference Dedicated to Professional Image Chains (Image'Com 90)
- * T. Lasey, B. Tuck, R. Beckwith, "An image server for interactive video network", 1st Int. Conference Dedicated to Professional Image Chains (Image'Com 90)
- * R. Castanet, X. Navarro, "ITAM: un service de transmission d'images numériques", 1st Int. Conference Dedicated to Professional Image Chains (Image'Com 90)