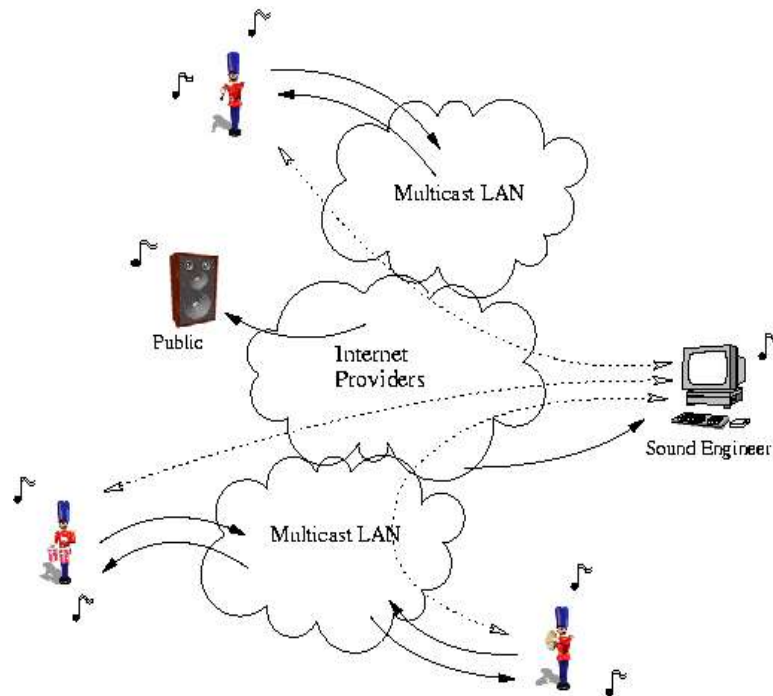
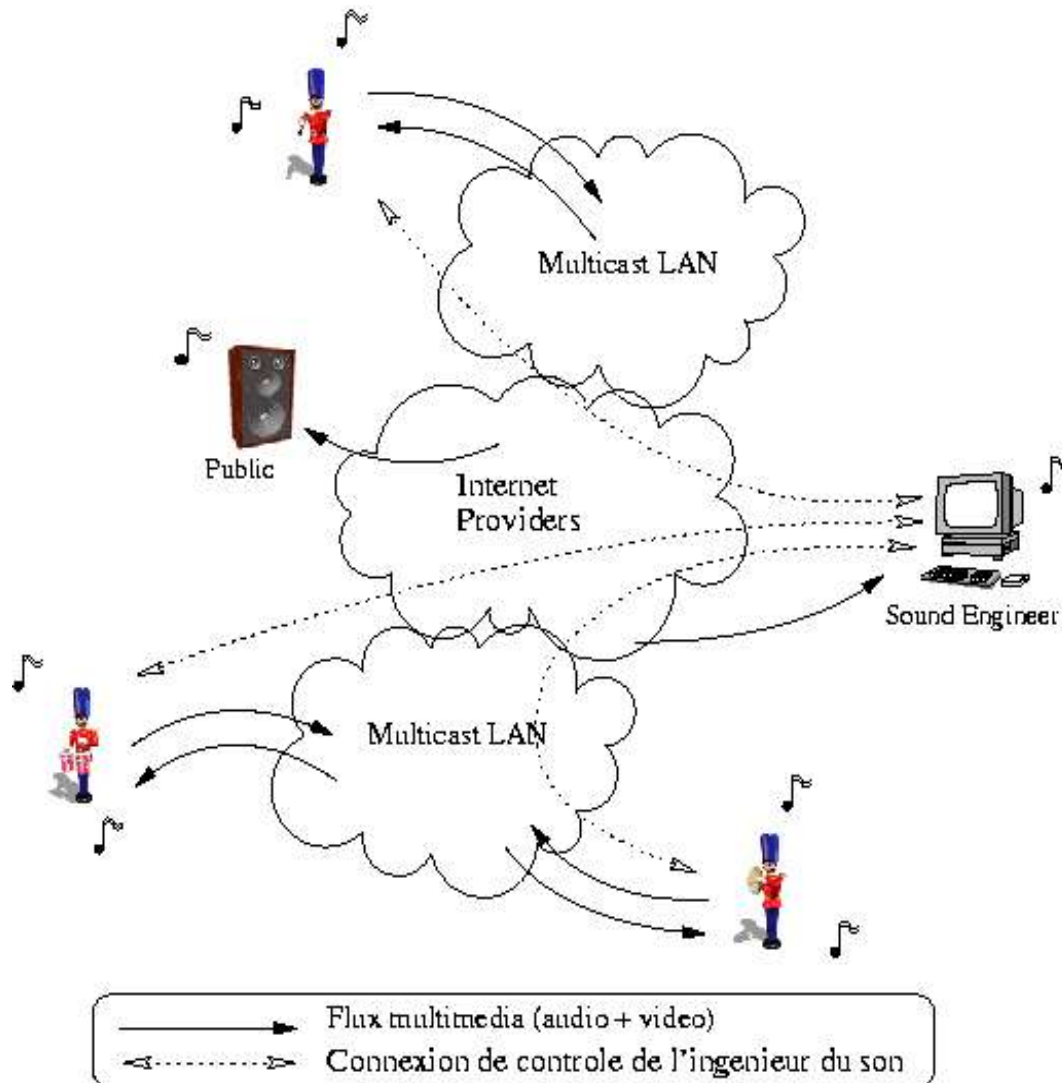


Le concert virtuel réparti sur l'Internet : vers une approche composant



Nicolas Bouillot
Hans-Nikolas Locher

Le concert Virtuel Réparti : objectifs



- Les musiciens s'entendent et se voient mutuellement
- Interaction entre musiciens
- Contrôle des périphériques audio des musiciens par un ingénieur du son
- Le public peut écouter le concert en « Live » sur le réseau (à la maison ou dans un auditorium)

Présentation du concert réparti

Présentation du concert réparti

Echange de flux

Supervision

Architecture actuelle

Vers une approche composants

Sémantique des échanges

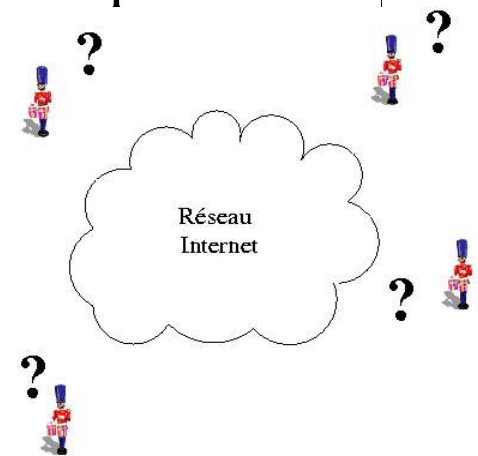
et communication locale

L'ordonnancement des composants

Conclusion

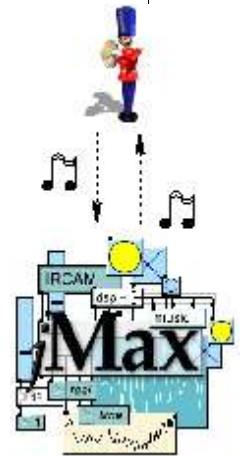
La communication audio entre musiciens : des choix à faire

- Les musiciens doivent tous s'entendre mutuellement
 - Notion de groupe
- Un objectif : l'interactivité
 - Contrôle de la cohérence et des délais de bout en bout
- Qualité: nécessaire pour la diffusion du concert vers un public
 - PCM 44100Hz multi-canaux (format CD)
- *Quel schéma de communication ?*



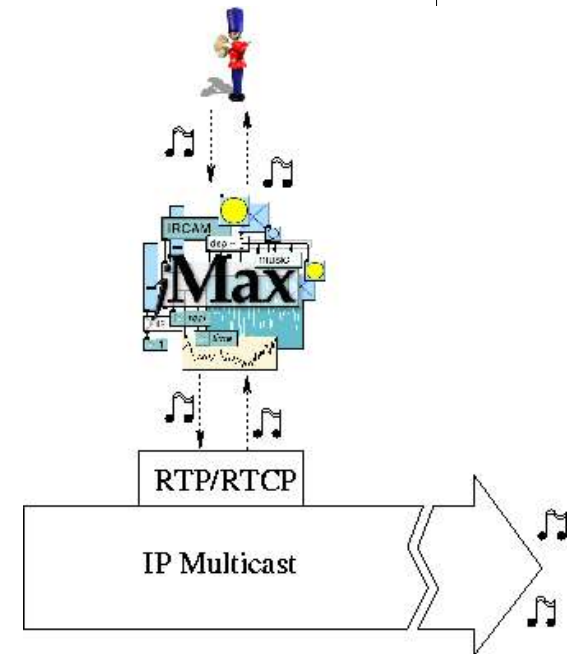
Acquisition et restitution du son

- Besoin d'un outil d'intégration :
 - Acquisition/restitution du son
 - Interface graphique (paramétrage par l'utilisateur)
 - Glue vers la partie transport réseau
- Jmax: un outil de conception d'applications multimédia temps réel
 - Plateforme orientée objet
 - Notion de flux + plugins



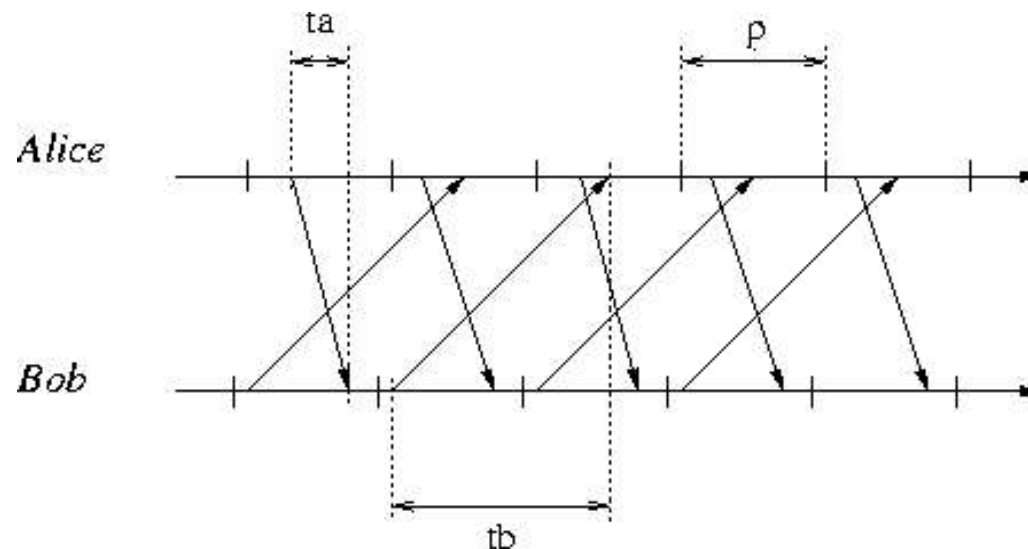
Transport réseau des flux audio

- Contraintes de délais
 - Pas d'acquittement (fiabilité ?)
 - Délivrance (estampillage logique)
- Notion de groupe
 - Diffusion du flux
 - Entrée et départ au sein du groupe
 - *RTP/RTCP au dessus d'IP multicast*



La communication par flux

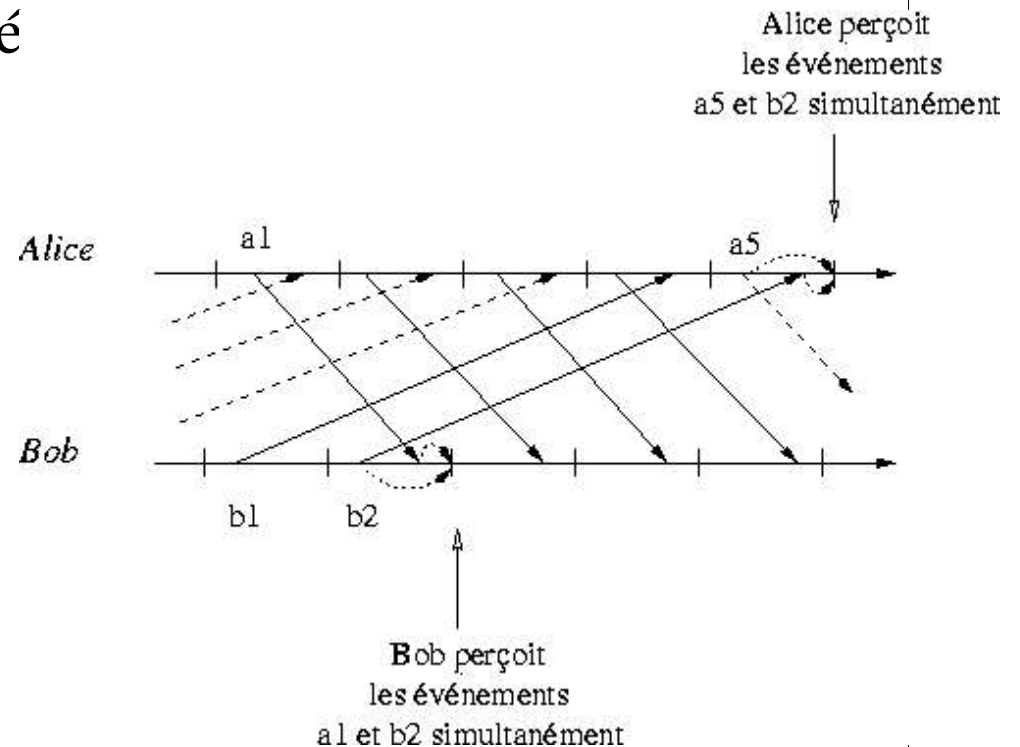
- Acquisition, restitution, transport effectués en temps réel



- T_a , T_b et p sont des valeurs constantes

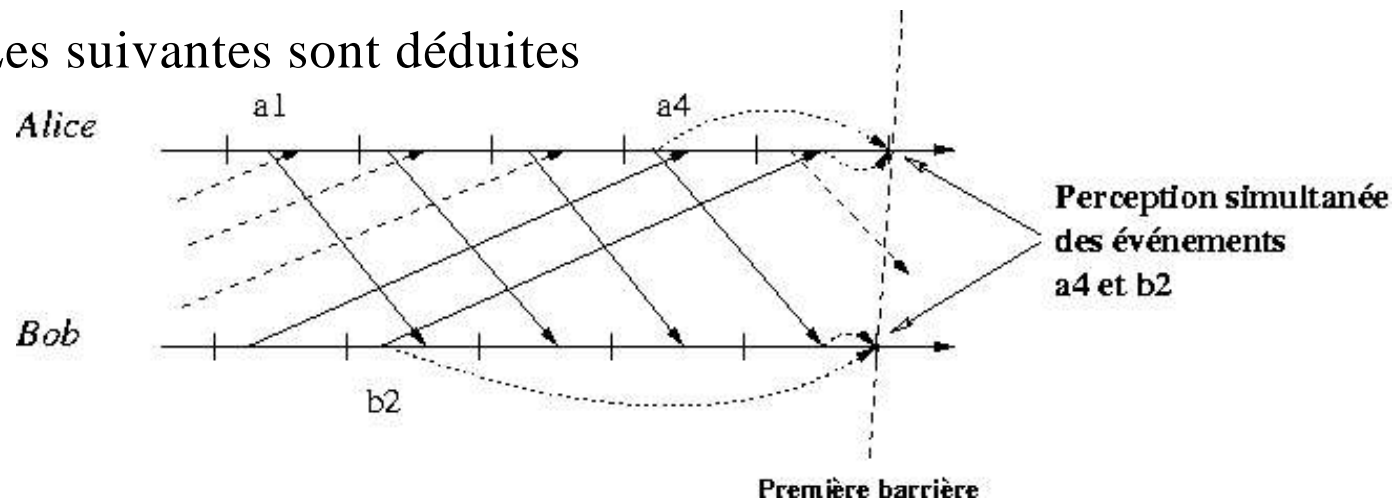
La cohérence perceptive

- La communication par flux est insuffisante pour l'interactivité
- En salle, les musiciens s'entendent et se voient simultanément
 - Cohérence perceptive
- *Synchronisation à l'aide de l'algorithmique répartie*



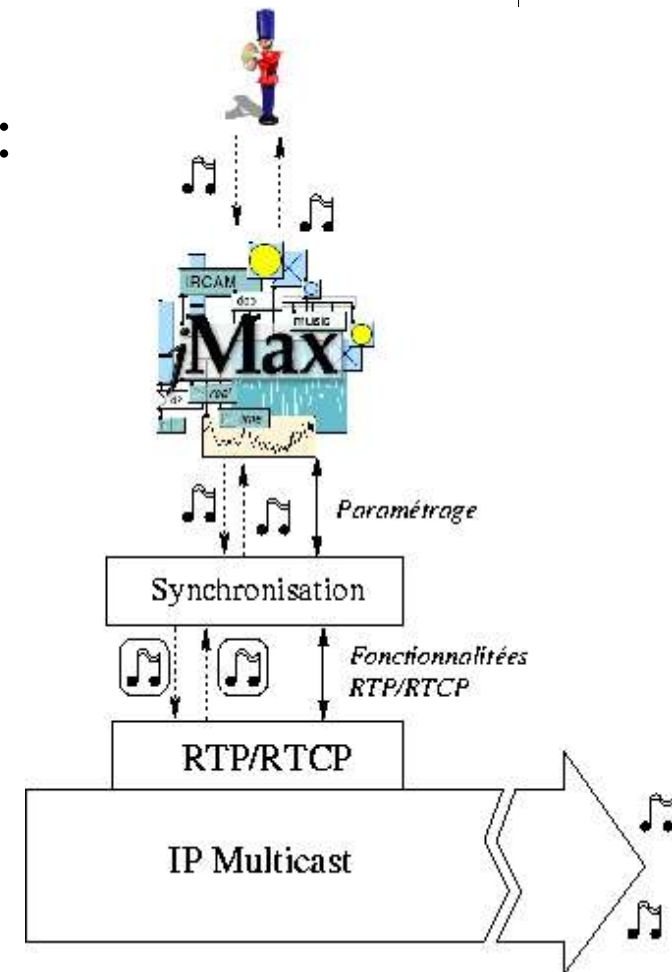
Principe de la synchronisation

- Le son local est retardé
- *Hypothèse : Les musiciens savent jouer avec un décalage, pourvu qu'il soit constant*
- Synchronisation de type barrière:
 - La première est déterminée par algorithmique répartie
 - Les suivantes sont déduites



La cohérence perceptive dans le système

- La synchronisation du mixage des flux se fait avant la restitution aux musiciens:
 - Optimisation du délais local ajouté
 - Estampillage véhiculé par RTP
 - Paramétrage par l'utilisateur, s'il le souhaite (augmentation du délai global)
- Les couches de synchronisation et transport communiquent par flux de données structurées



Interactivité et multimédia distribué ? (1/2)

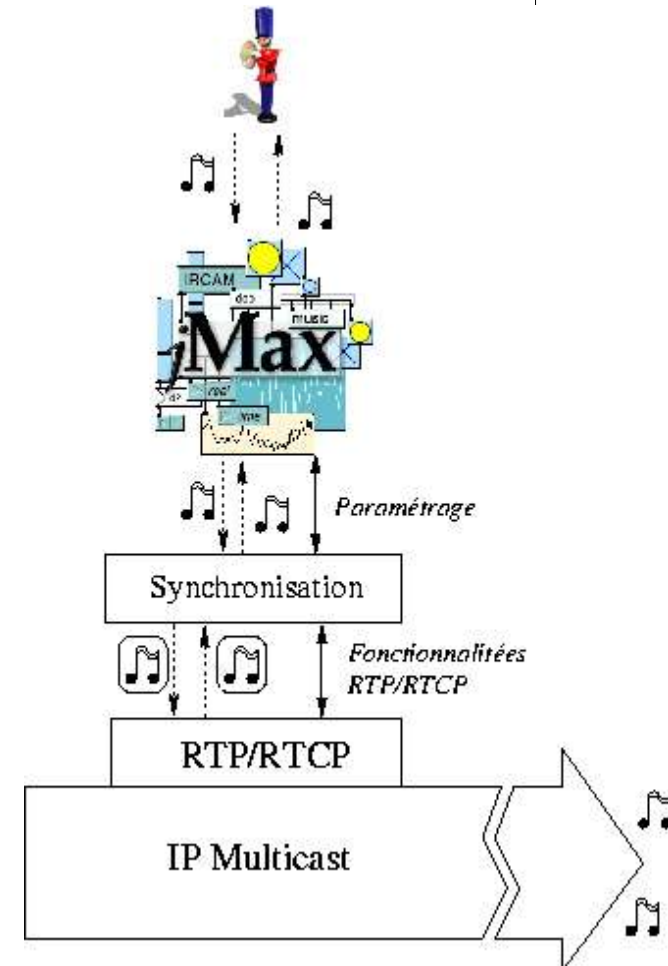
- Applications à collaborations autonomes
 - Les efforts des participants sont parallèles et concurrents
 - *Réalité virtuelle, Jeux, Tableau blanc, Gestion d'emploi du temps*
- Applications à collaborations synchrones
 - Les efforts sont simultanés (pas de concurrence)
 - *Théâtre virtuel, Réalité virtuelle, concert réparti*

Interactivité et multimédia distribué ? (2/2)

- Univers à collaboration autonome
 - Ordre total global sur les événements + contraintes d'échéances
- Univers à collaboration synchrone
 - Cohérence perceptive + contraintes d'échéances
- *L'univers virtuel doit contenir des mécanismes permettant le(s) type(s) de collaboration(s)*
- *La synchronisation est applicable aux applications à collaborations synchrones*

La configuration par l'utilisateur

- Joindre ou quitter le groupe
- Choisir un retard plus grand que celui imposé par la synchronisation
 - Une mesure, deux temps...
- Métronome
 - Activer
 - Choisir le pas (nombre de battements par temps)



Présentation du concert réparti

Présentation du concert réparti

Echanges de flux

Supervision

Architecture actuelle

Vers une approche composants

Sémantique des échanges

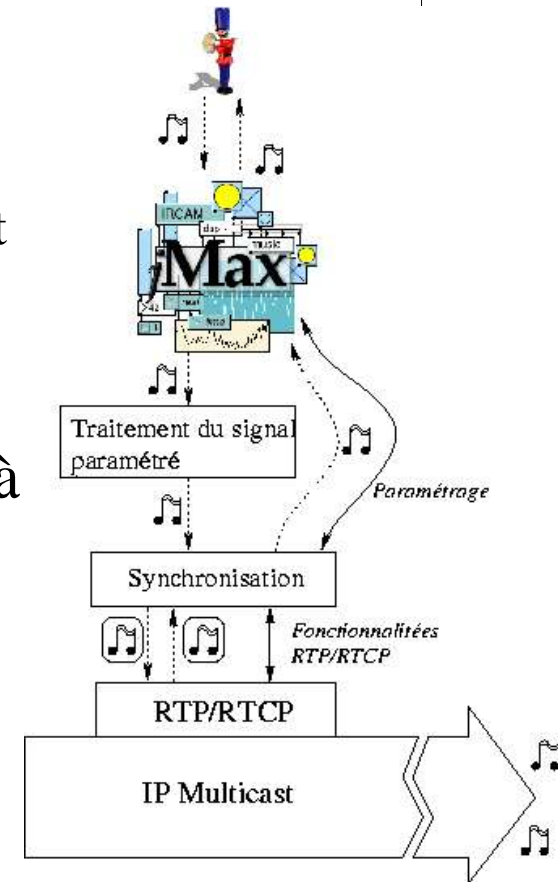
et communication locale

L'ordonnancement des composants

Conclusion

Besoin de supervision

- Console pour l'ingénieur du son
 - Paramètres à régler :
 - Paramètres de traitement du signal (effectué avant émission du son).
 - Arrêt et démarrage des flux.
 - Doit être déportée pour permettre un contrôle à distance
 - La vue des paramètres doit être tenue à jour
- Eventuellement, paramétrage de l'initialisation du système



Service de distribution de données

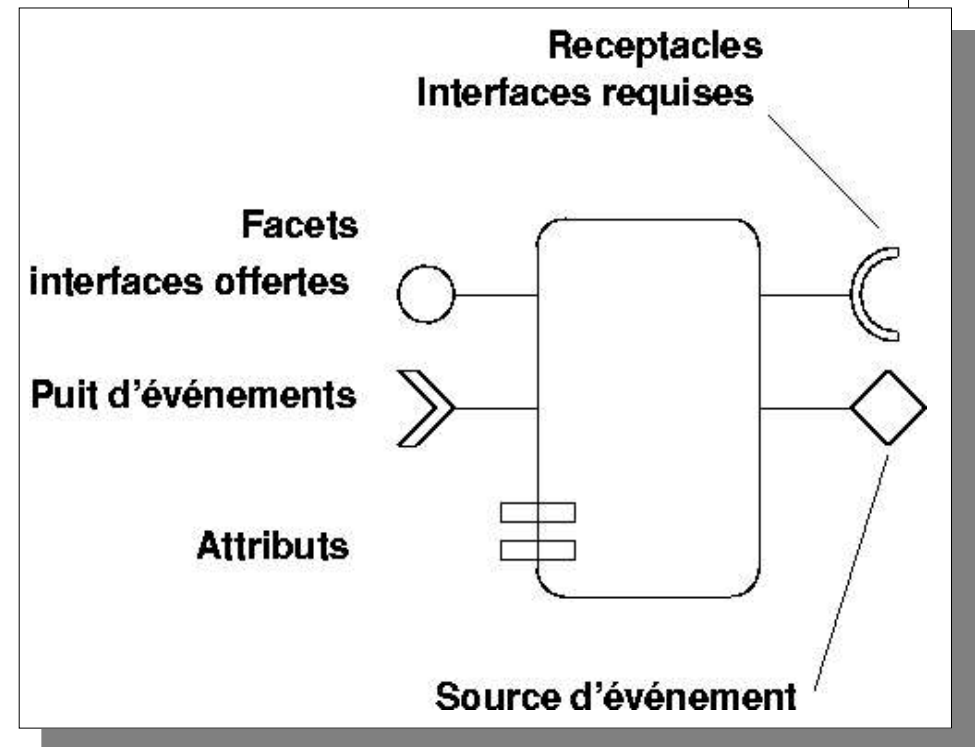
- Le serveur
 - propose une liste de variables, et la possibilité d'y souscrire soit individuellement, soit par ensemble.
 - Notifie le client en transportant la valeur de la donnée
 - Au changements des valeurs, avec ou sans mise en tampon
 - Périodiquement
 - Sur requête du superviseur
- Le client
 - Consulte les listes de données, et constitue les ensembles de données, positionne la politique de notification.

Le logiciel OpenTAZ

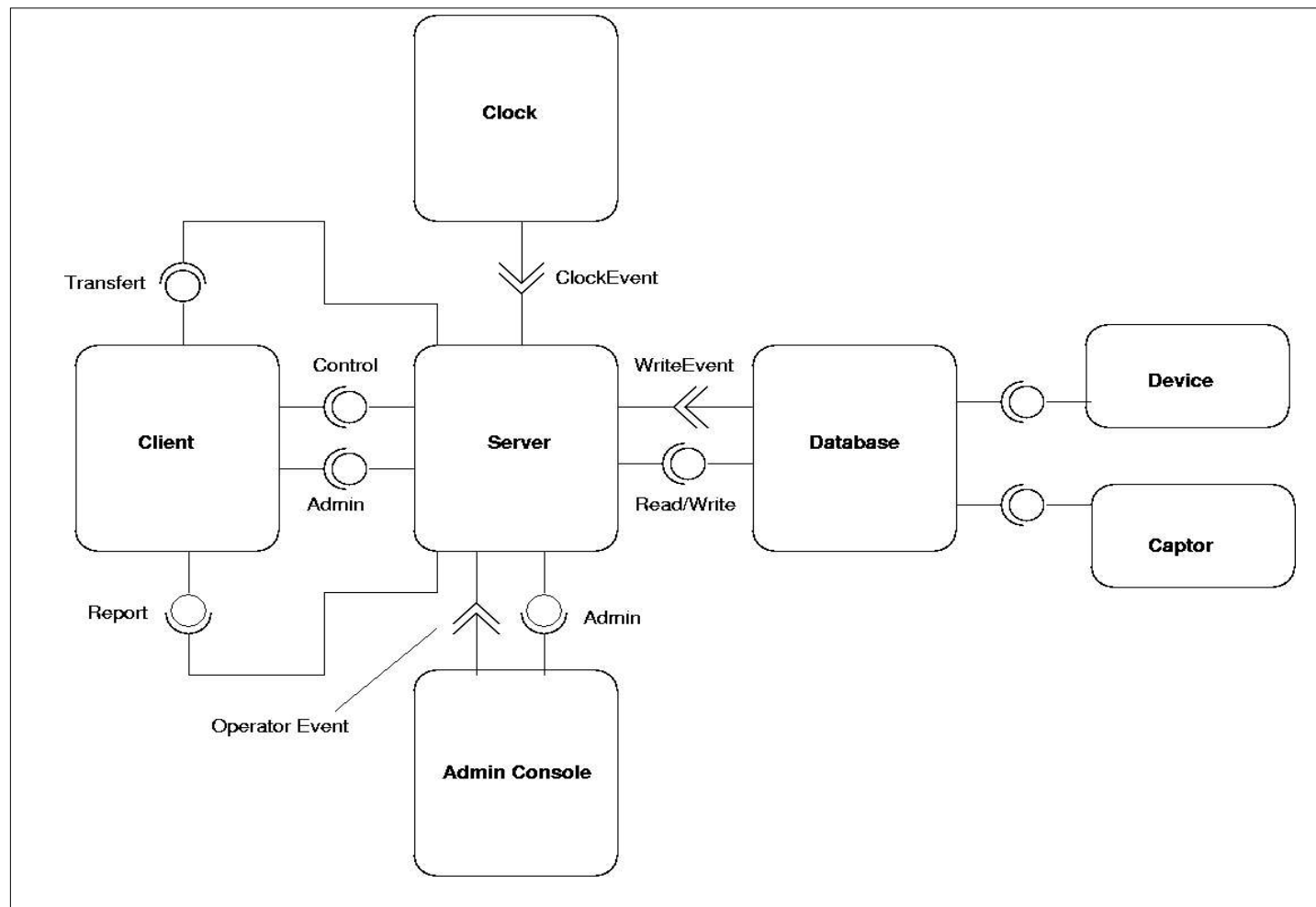
- Conforme aux spécifications de la norme IEC-TASE.2, pour la supervision centrale de production d'énergie
- La partie serveur met en correspondance un périphérique réel et un jeu de paramètres publiés.
- Transposition des concepts au-dessus de CORBA.
- Implémentation en C++ avec Mico 2.3.6.
- Travail de thèse de Erwan Bequet

Le modèle de composants CORBA (CCM)

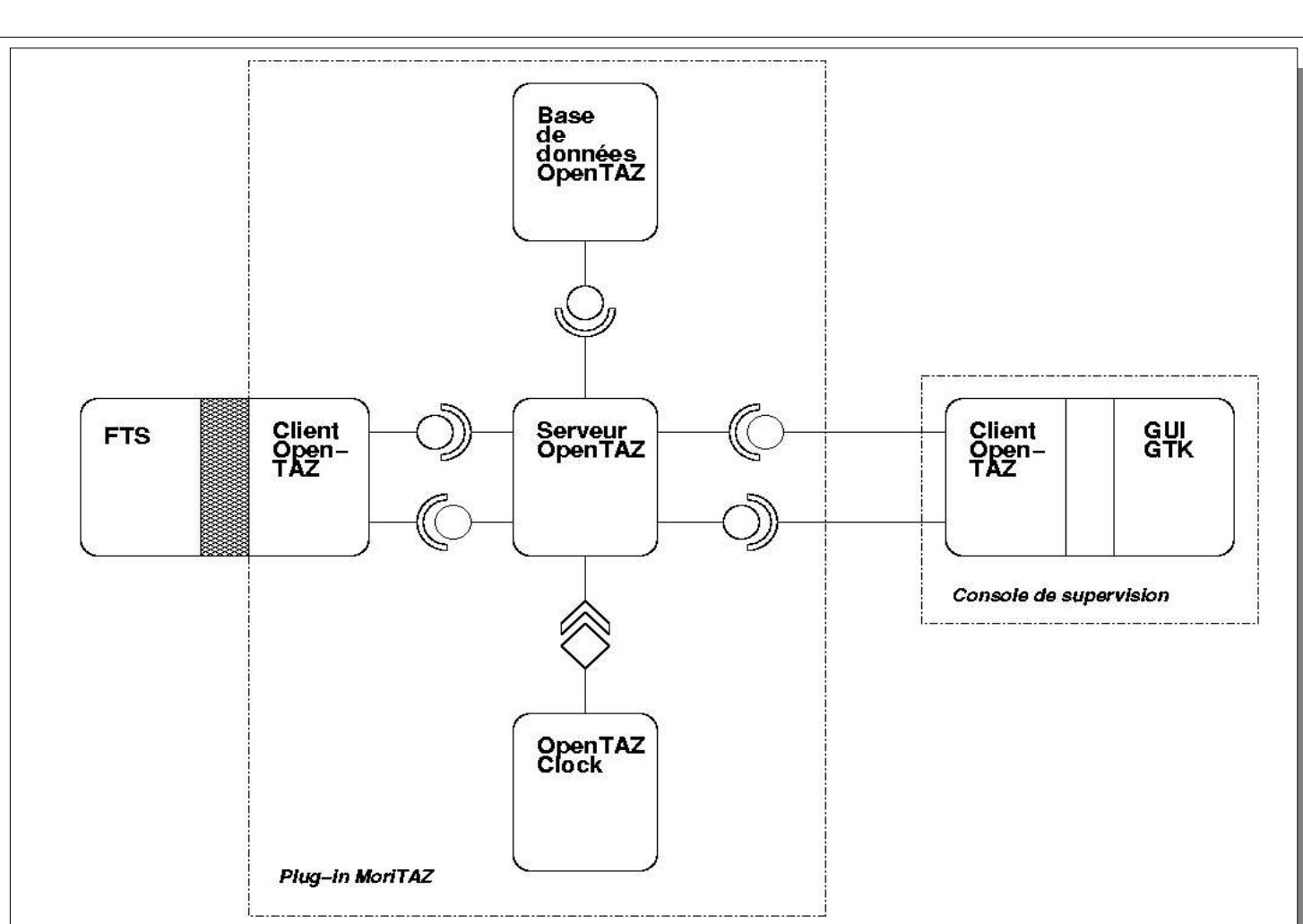
- Une architecture normalisée par l'OMG
- Specification
 - Interfaces
 - Empaquetage
 - Déploiement
 - etc...



Une première experience avec CCM : OpenTAZ-CCM



Architecture de la supervision



OpenTAZ-CCM : quelques retours d'expérience

- La modularité
 - Sous-exploitée dans le prototype actuel
 - Hétérogénéité avec les autres éléments de l'application
 - Exemple : l'horloge pourrait être réutilisée
- La charge du système est importante
 - Supérieure à celle constatée avec la version non-CCM !
 - Parasitage de la gestion des flux sur la même machine
 - Exprime le manque de priorité de l'échange de flux par rapport aux données de contrôle ?
- La gestion des ressources est confuse

Présentation du concert réparti

Présentation du concert réparti

Echanges de flux

Supervision

Architecture actuelle

Vers une approche composants

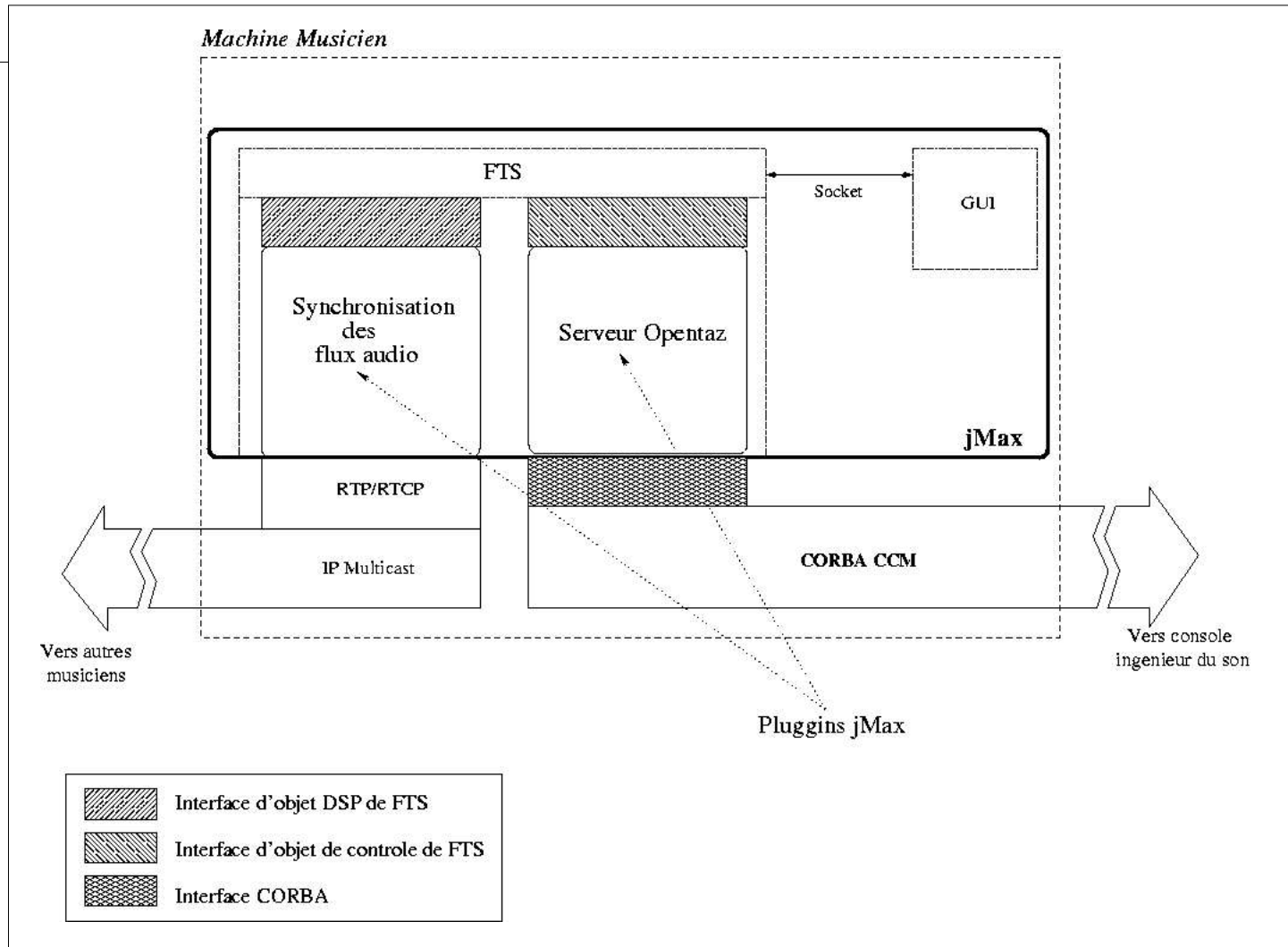
Sémantique des échanges

et communication locale

L'ordonnancement des composants

Conclusion

Architecture de l'application



Vers une architecture composant

Présentation du concert réparti

Echanges de flux

Supervision

Architecture actuelle

Vers une approche composants

Sémantique des échanges

et communication locale

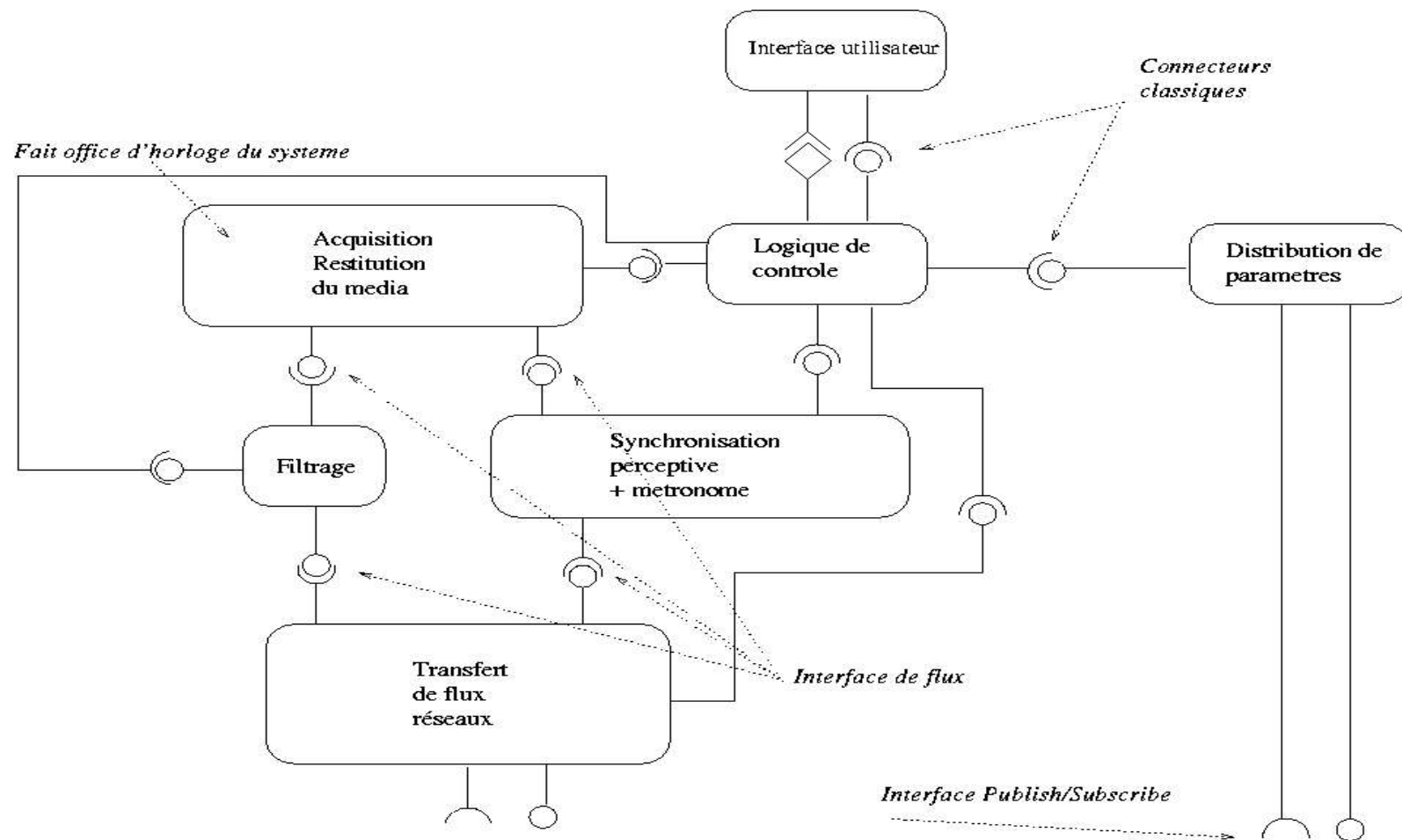
L'ordonnancement des composants

Conclusion

Prospective

- Mécanismes réutilisables présents dans le système :
 - Transfert de flux audio
 - Synchronisation
 - Acquisition et restitution du son
 - Data Centric Publish/Subscribe
- *Comment intégrer ces mécanismes dans l'architecture CORBA CCM ?*

Une architecture CCM possible



Vers une architecture composant

Présentation du concert réparti

Echange de flux

Supervision

Architecture actuelle

Vers une approche composants

**Sémantique des échanges
et communication locale**

L'ordonnancement des composants

Conclusion

Sémantique des échanges

- Interfaces CCM
 - Appel de méthodes
 - Notification d'événements
- *Echanges non-prédéfinis*
 - *Flux*
 - Service de transfert de flux CORBA/Streaming
 - Besoin d'une interface locale (performances).
 - *Data Centric Publish/Subscribe*
 - Sémantique de mémoire partagée répartie.

Echange des flux :

Service CORBA/Streaming

- L'architecture intègre :
 - Interfaces avec sémantiques d'établissement et de contrôle des flux
 - Encapsulation de protocoles de transferts de données (TCP-UDP-RTP)
- L'architecture rend possible :
 - La transparence à la localisation
 - Le traitement particulier des flux par l'application (pour la synchronisation, l'accès au timecode...)
 - Le paramétrage de la QoS (système et réseau)

Echange des flux :

Service CORBA/Streaming

- Les données ne passent pas par l'IIOP
- Réification des données protocolaires à l'application
 - Configuration lors de l'établissement
 - Traitement (Réception de « Frame » avec « Frame_info » associée par callback)
- Implémentations
 - TAO (via ACE) pour le streaming
 - Quartz (prototype), QuO (avec TAO) pour la QoS
- Service orienté transport réseau

Data Centric Publish/Suscribe

- Sémantique de mémoire partagée répartie
 - Mode d'échange plus efficace que la mémoire paginée répartie
- Fonctionne d'une façon proche des services d'événements
 - Le Publisher publie des données hierarchiques
 - Un Subscriber s'abonne aux données qui l'interesse
 - Le Publisher notifie les changements dans l'arbre de données, et la modification des données souscrites

CORBA

et la communication locale

- FTS : objets dans le même espace mémoire. Segments de flux échangés par référence
- CORBA : copies multiples pour la communication entre les composants
- Optimisation envisageable des communications locales
 - Utilisation d'un adaptateur pour éviter le passage dans l'ORB (et le POA).
 - Pas d'implémentation disponibles à notre connaissance

Vers une architecture composant

Présentation du concert réparti

Echange de flux

Supervision

Architecture actuelle

Vers une approche composants

Sémantique des échanges

et communication locale

L'ordonnement des composants

Conclusion

Ordonnancement : le modèle de FTS

- A chaque impulsion d'horloge, chaque flux avance d'un segment.
 - Les objets sont organisés dans un graphe sans cycle
 - La relation Producteur-Consommateur oriente les arcs
 - Garantie l'intégrité des données (non recouvrement des tampons)
 - Les tampons sont pré-alloués et évitent les appels systèmes
- Approche synchrone (Lustre, Signal, Esterel)
- Quelles garanties dans CORBA ?

Ordonnancement des invocations dans CORBA/RT

- Priorité sur les invocations
 - Fixées côté serveur
 - Propagées côté client
- Les priorité sont projetées sur celles du système d'exploitation
 - Support du système sous-jacent nécessaire
- Implémentation du *Scheduling Service* dans TAO

Conclusion

Présentation du concert réparti

Echanges de flux

Supervision

Architecture actuelle

Vers une approche composants

Sémantique des échanges

et communication locale

L'ordonnancement des composants

Conclusion

Conclusion

- Plateforme FTS
 - Sémantique d'échange entre objets adaptée au multimedia (composition, hierarchisation, flux...)
 - N'est pas un standard
 - Objet locaux uniquement
- Plateforme CCM
 - Standard
 - Interfaces génériques pour la communication entre objets
 - Pas d'optimisations locales