

RIS



**Point sur l'utilisation des technologies
middleware dans les centres de contrôle et de
mission des systèmes spatiaux**

P. Rouzet, F. Lecouat

Le choix du middleware CORBA

- **Émergence des applications distribuées :**
 - architectures réparties
 - communications inter-process
- **Plusieurs possibilités :**
 - socket
 - RPC
 - middleware "maison" ou propriétaire (Ilog Broker)
 - CORBA
- **Les intérêts de CORBA :**
 - formalisme objet adapté au langages utilisés (C++, Java)
 - aspect normé (OMG)
 - abstraction des couches basses (développement plus rapide)
 - support multi-plateformes, fournisseurs multiples (dépendance)
 - sensibilité du client à la technologie utilisée

État des lieux

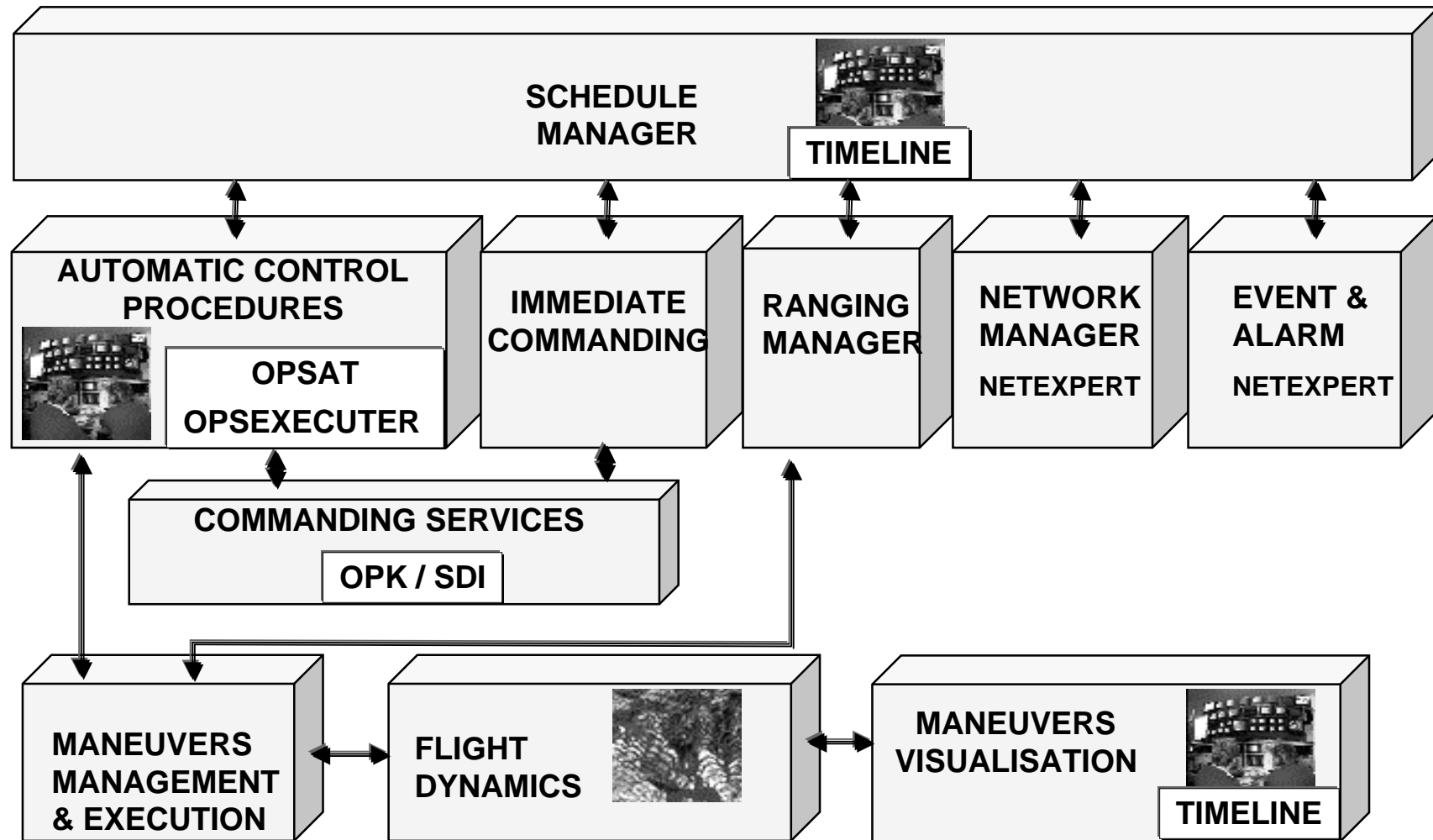
- **Open Center**

- ensemble de composants logiciels à la fois techniques (framework) et métiers permettant la réalisation de :
 - centres de contrôles (Intelsat FDC)
 - bancs de test (Essaim, Rocsat 2, Metop ...)
 - simulateurs satellites (Intelsat X, Inmarsat 4 ...)

- **OpsWare**

- suite de produits logiciels orientés vers l'automatisation des opérations satellite (planification et exécution de procédures d'opération) venant en surcouche de fonctions de base :
 - centres de contrôles (Intelsat FDC, SCC E3000 ...)
 - simulateurs satellites (Intelsat X, Inmarsat 4 ...)
 - centres de mission (séquenceur CSU HII)

Exemples d'application (1) – Intelsat FDC



Exemples d'application (2) – Intelsat FDC

- **Operations**

- System designed for 30 satellites from 7 series, 2 control centers, 6 earth stations and 30 simultaneous users
- Scheduling : 30 days in advance and 7 days in the past (9000 sessions)
- Commanding reactivity : 15 seconds
- Network management : 48 antennas, 1500 devices

- **Availability**

- MTBF 3000h, restart in less than 3 minutes
- Cluster architecture
- "ORB Monitoring & Control"
 - sockets management
 - daemon interactions

Exemples d'application (3) – E3000 SCC

- **Centre de contrôle pour la nouvelle génération de satellites Astrium (plate-forme Eurostar 3000)**
 - => intégration avec des composants tiers
 - couche opérations OpsWare
 - noyau TM/TC et framework Inmarsat/L3-Storm
 - gestion de la tolérance aux pannes

- **Nouvelle version du produit OpsExecuter (suite OpsWare) intégrant une IHM distribuée Java**
 - => langages et plate-formes hétérogènes
 - noyau C++ avec broker Orbix sur Solaris
 - IHM Java avec broker jdk 1.3 sur Solaris ou Windows NT
 - gestion de la tolérance aux pannes

Points positifs

- **Simplification de l'implémentation de la communication inter-process**
- **Uniformisation des définitions et des mécanismes mis en jeu dans les interfaces (communication inter-équipes)**
- **Utilisation assez large permettant l'intégration rapide de produits tiers (analyse des interfaces facilitée)**
- **Déploiement des process relativement transparent**
- **Interopérabilité des ORB (IIOP)**
- **Services associés (event, transaction, naming ...)**
- **Versatilité des modèles d'invocation d'interface (threads dédiés ...)**

Points négatifs (1)

- **Concepts parfois difficiles à maîtriser lorsqu'on va au fond du formalisme objet**
- **Maîtrise de l'implémentation (gestion mémoire en particulier – compteurs de références, paramètres out, sequences ...)**
- **Mise en place de la tolérance aux pannes (ouverture réseau, perte machine)**
- **Tentation d'utiliser des aspects non standards fournis par un ORB particulier**
- **Évolution des ORB et de la norme (BOAImpl / POA)**

Points négatifs (2)

- **Coût des licences**
- **Support technique pas toujours efficace**
- **Bugs dans les ORB (fuites mémoires daemon)**
- **Disponibilité en fonction des plate-formes et des versions des OS**

Les tendances actuelles

- Revenir à du "CORBA compliant"
- Utilisation d'ORB free (OmniORB, TAO ...)
- Prise en compte accrue de la gestion de la tolérance au pannes (logiciel, matériel)
- Problématique du "Naming Service" (interopérabilité, reprise sur contexte)