Couche 7/OSI : SERVICE DE MESSAGERIE INDUSTRIELLE MMS
Manufacturing Message Specification

Ce service Application constitue la base des standards MAP (Manufacturing Automation Protocol). Les fonctions qu'il fournit sont directement accessibles à l'utilisateur. Il est défini dans le standard OSI 9506. La partie 9506-1 spécifie le service et la partie 9506-2 le protocole.

Ce service est basé sur un mécanisme **client-serveur** : des systèmes "serveurs" (automates, robots, manipulateurs, etc.) sont utilisés à la demande par les "clients" que sont les systèmes de supervision ou d'autres automates. Un équipement peut être à la fois client et serveur. L'architecture n'est pas purement client-serveur; elle permet la notification d'événements au client par le serveur.

En pratique, il est commode de le structurer en deux sous-couches : Un service "temps-réel" qui fournit les fonctionnalités prévues et un service "communication ou messagerie) chargé de transmettre les PDU entre les entités qui coopèrent et éventuellement d'annuler des demandes en cours.

5.1. SERVICE REQUIS

MMS ne contient aucun mécanisme d'adressage et s'appuie sur une association entre entités d'applications établie par le service ACSE . Après établissement de cette association les PDU sont transférés directement sur la connexion de Présentation correspondante.

Nota : Les fonctionnalités de ce service pourraient dont être aussi mises en oeuvre dans un système de communication non-OSI qui fournirait simplement un mécanisme de transfert de blocs de données bidirectionnel à l'alternat et un mécanisme d'adressage.

5.2. SERVICE FOURNI

Le service MMS offre une très grande variété de fonctions: 84 services élémentaires et 2 modalités, qui sont regroupées logiquement en 9 sous-ensembles pour en faciliter la description. Ce sont :

Gestion de contexte
Gestion de l'Equipement Virtuel de Production
Gestion de variables
Gestion de programmes
Gestion d'événements
Communication opérateur
Gestion de domaines
Gestion de sémaphores
Gestion de journal

et un sous- ensemble annexe :
Gestion de fichiers.

Deux modalités complémentaires peuvent compléter tout service en le liant à un sémaphore ou à une condition sur événement.

De nombreux projets, par exemple le profil européen CNMA, prévoient de n'implanter dans un premier temps qu'un sous-ensemble des 84 fonctions, utilisées le plus fréquemment dans les équipements industriels actuels : automates, robots, systèmes de convoyage.

Nous ne pouvons détailler ici toutes les fonctions qui ne seront parfois qu' énumérées. Les noms des fonctions sont en général très explicites et indiquent l'objet de celles-ci. Une description de chaque fonction est donnée dans les spécifications d'interface.

5.3. SOUS-ENSEMBLES FONCTIONNELS

5.3.1. Modalités

Elles permettent de conditionner l'exécution des fonctions ci-dessous à la disposition d'un sémaphore ou à la réalisation d'une condition liée à un événement :

Ce sont :

attachToSemaphore
attachToEventCondition

5.3.2. Gestion de contexte

La Gestion de Contexte permet d'**initialiser et de terminer** un dialogue entre entités MMS paires et d'en **négocier les options**. Elle utilise le service ACSE. Les options négociables portent en particulier sur la liste des services élémentaires à mettre en oeuvre et les paramètres utilisés.

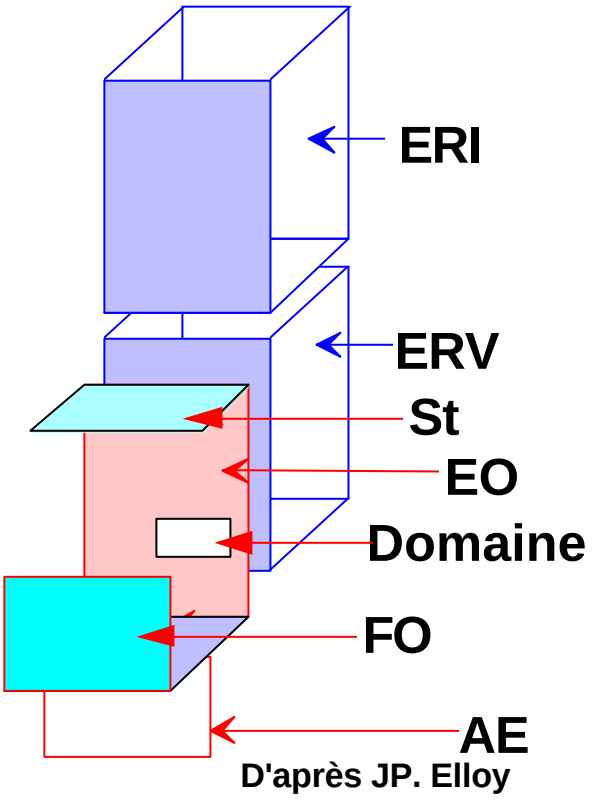
primitives utilisées :	initiate	initialisation
	conclude	terminaison normale
	cancel	annulation du service
	abort	terminaison sur anomalie
	reject	refus du service

Ce sous-ensemble doit obligatoirement être implanté.

5.3.3. Gestion de l'Equipement Virtuel de Production

La notion d'Equipement Virtuel de Production (EVP ou VMD Virtual Manufacturing Device) est **très importante**. Elle **ne s'applique qu'aux serveurs MMS**. Coté client, une interface permettant de demander l'exécution de services est en cours de spécification (MMS Application Interface Specification) .

L'équipement virtuel de production permet de définir de manière logique un appareil (automate, robot, commande numérique de machine outil, etc.) ou une fonction de l'**équipement réel de production (ERP)**. L'EVP comporte un ensemble **d'objets visibles,** objets logiques, représentés dans la mémoire de l'équipement réel (ou d'un système frontal pour des équipements réels trop peu puissants) et mis en correspondance avec des objets réels par un logiciel spécifique, créé par l'équipementier) (par exemple une procédure de conversion).



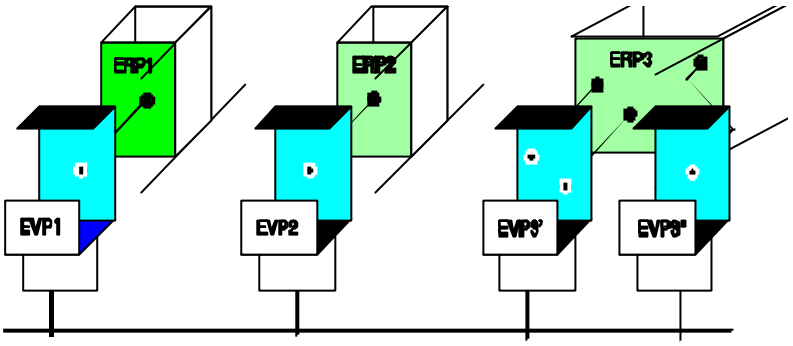
Certaines parties de l'équipement réel peuvent ne pas être traitées dans l'Equipement Virtuel et n'avoir qu'une portée locale. Elles sont notées ERI sur le schéma ci-contre. La partie visible de l'ERP est notée ERV.

Ainsi la communication n'est pas établie directement avec l'appareil réel, mais à travers une sorte de "sas" dans lequel MMS dépose ses

requêtes (commandes, lecture de variables, etc.) ou vient lire les informations dont a besoin l'autre application.

L'appareil réel met à jour les "objets visibles" (variables,... notés EO) partagés ou vient retirer les commandes dans ce sas; la communication est ainsi désynchronisée du fonctionnement du processus réel.

Le schéma ci-dessous montre des exemples de correspondance entre équipement réels de production (partie "visible") et équipement virtuel correspondant. Cette correspondance n'est pas forcément biunivoque et un équipement réel complexe peut être "représenté" par plusieurs EVP correspondant à des groupes de fonctionnalités.



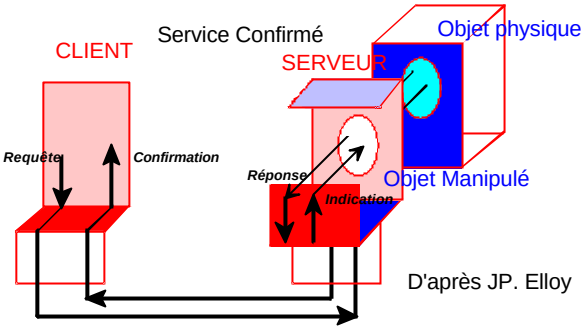
Les objets de l'EVP sont répartis en une dizaine de classes:

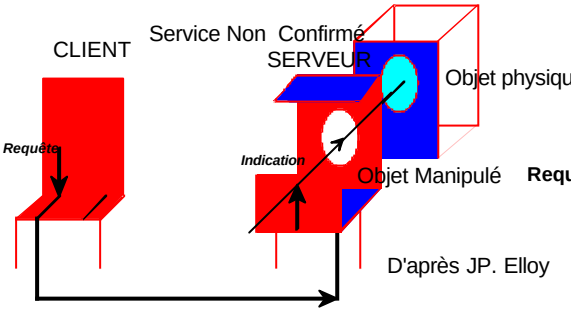
- Variable
- Type
- Sémaphore
- Condition événementielle
- Action événementielle
- Enveloppe événementielle
- Journal
- Domaine
- Invocation Programme
- Station opérateur

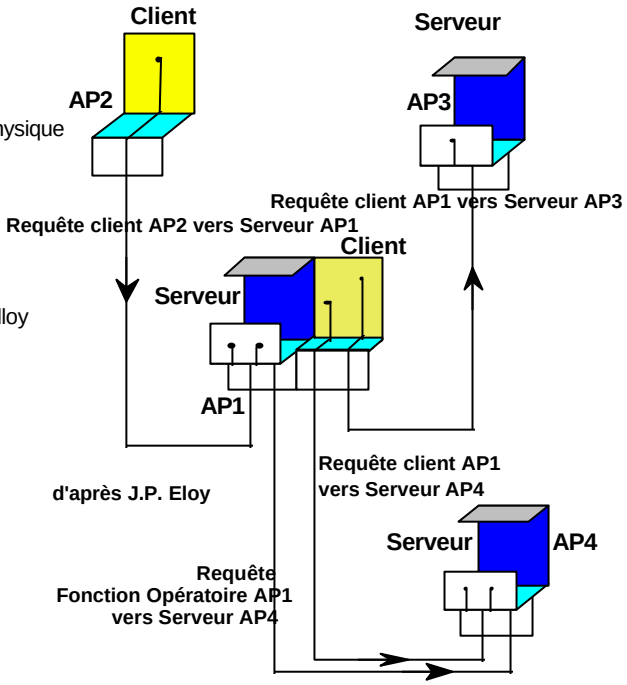
Un **domaine** est un regroupement d'objets et/ou un ensemble d'informations qui peuvent être **télé-chargées** et **télé-sauvegardées**.

Ces objets, qui sont mis en correspondance avec des ressources de l'équipement réel, sont traités par la **Fonction Opératoire** (notée FO sur le schéma ci-dessus). Cette Fonction Opératoire réalise l'accès (en lecture ou en écriture) aux objets de l'EVP sur une demande de service de la part du système client ou sur des requêtes du système local.

Le service est la plupart du temps confirmé. Le schéma ci-contre illustre le mécanisme qui en permet la mise en oeuvre.







Quelques services ne sont pas confirmés (eventNotification, abort, reject, informationReport, unsolli-citedStatus). Le mécanisme mis en oeuvre est plus simple.

Le schéma ci-contre illustre une coopération plus complexe entre quatre systèmes : un client, deux serveurs et un système client-serveur mettant en oeuvre des services confirmées ou non-confirmés.

Les noms de ces objets ont une portée variable :

Portée EVP
Portée Domaine (sous-ensemble de l'EVP)
Portée Association d'Applications (ensemble client-serveur associés pour traiter une application)

Tous les classes peuvent avoir l'EVP comme portée. Le journal, un domaine, une invocation de programme ou la station opérateur ne peuvent être réduit à un domaine. Un sémaphore, de même qu'un domaine, une invocation de programme ou une station opérateur, ne peuvent avoir une portée dépassant l'EVP.

Primitives utilisées : status
getNameList
identify
rename
getCapabilityList
informationReport
unsollicitedStatus

5.3.4. Gestion de variables

Une variable correspond à un ou plusieurs éléments de données qui sont référencées pour le partenaire par un nom ou une description . Ces variables peuvent être prédéfinies par l'équipement réel; elles ne peuvent alors pas être renommées ou supprimées.

Un certain nombre de services élémentaires de ce sous-ensemble est nécessaire dans tous les types d'équipements réels:

La gestion de variables permet ainsi d'envoyer des commandes à un équipement réel, robot ou automate par exemple, ou de lire des informations sur un appareil. Ces variables peuvent être repérées par leur emplacement (adresse mémoire) comme dans les systèmes réels courants mais aussi par un nom logique.

Ce sous-ensemble permet en particulier :
- d'assigner un nom symbolique à un type de données
- d'obtenir le type associé à un nom symbolique
- de dissocier nom symbolique et type

Primitives associées : defineNamedType
getNamedTypeAttibutes
deleteNamedType

- d'assigner un nom symbolique à un emplacement distant
- d'obtenir l'emplacement associé à un nom symbolique

- de dissocier nom symbolique et emplacement

Primitives associées : defineNamedVariable
getVariableAccesAttributes
deleteVariableAcces

- de lire ou de changer le contenu de une ou plusieurs variables d'une entité paire
- d'envoyer des informations à l'entité paire sans qu'elles soient demandées.

Primitives utilisées : read
write

- de manipuler des listes de variables

Primitives utilisées : defineNamedVariableList
getNamedVariablesList
AttibutesdeleteNamedVariableList

Autres primitives : defineScatteredAcces
getScatteredAccesAttributes
deleteVariableAcces

5.3.5. Gestion de programmes

La gestion de programmes, associées à des fonctions de gestion de fichiers, permet le **lancement** de programmes ou leur arrêt. Il permet aussi de les suspendre ou de les relancer.

Leur **téléchargement** ou leur **sauvegarde** ressort de la **gestion de domaines** ou des fonctions annexes de transfert de fichiers.

Primitives associées : createProgramInvocation
deleteProgramInvocation
start
stop
resume
reset
kill
getProgramInvocationAttributes

5.3.6. Gestion d'événements

Cette unité fonctionnelle permet de gérer des événements sur une machine distante en créant, manipulant, inspectant ou supprimant ces événements.

Il est possible de **tester une condition** liée à un événement, de **lancer une action** ou de **notifier une alarme sur un événement.**

Primitives utilisées :
defineEventCondition
alterEventConditionMonitoring
deleteEventCondition
getEventConditionAttributes
reportEventConditionStatus
triggerEvent
defineEventAction
deleteEventAction
getEventActionAttributes
reportEventActionStatus
defineEventEnrollment
deleteEventEnrollment
alterEventEnrollment
reportEventEnrollmentStatus
getEventEnrollmentStatus
getEventEnrollmentAttributes
acknowledgeEventNotification
getAlarmSummary
getAlarmEnrollmentSummary

eventNotification

Avec la gestion de sémaphores, la gestion d'événements fournit les fonctions de base pour la constitution d'un système temps réel distribué.

5.3.7. Gestion de sémaphores

Ce service permet, sur une machine distante, de définir ou de supprimer des sémaphores, de réserver, relâcher ou tester un sémaphore ou une ressource, enfin, de conditionner d'autres services selon la disponibilité d'une ressource (par un service modificateur) .

Primitives utilisées :
takeControl
relinquishControl
defineSemaphore
deleteSemaphore
reportSemaphoreStatus
reportPoolSemaphoreStatus
reportSemaphoreEntryStatus

5.3.8. Gestion de domaines

Ce sous-ensemble permet divers modes de **téléchargement de programmes** (ou de données) dans une zone mémoire spécifiée :

Chargement du local vers le distant (download)
Chargement du local depuis le distant (upload)
ou le stockage d'un programme sur la machine distante (load) .

Primitives utilisées :
initiateDownloadSequence
downLoadSegment
terminateDownloadSequence
initiateUploadSequence
uploadSegment
terminateUploadSequence
requestDomainDownLoad
requestDomainUpLoad
loadDomainContent
storeDomainContent
deleteDomain
getDomainAttibutes

5.3.9. Gestion de journal

Le sous-ensemble "Gestion de journal" permet de créer un **journal des événements** avec le contenu des variables associées et de lire ou écrire dans ce journal .

Primitives utilisées :
readJournal
writeJournal
initializeJournal
reportJournalStatus
createJournal
deleteJournal

5.3.10.Communication opérateur

Ce sous-ensemble permet de lire ou d'écrire sur les appareils d'entrée-sortie distants (console opérateur, clavier de commande, écran d'affichage, etc).

Primitives utilisées :
input
output

5.3.11. Annexe : Gestion de fichiers

MMS fournit un minimum de fonctions pour gérer et utiliser des fichiers. Pour un traitement complet on utilisera le service FTAM (File Transfert Acces and Management) qui fournit les primitives pour un système de gestion de fichiers distribué. (Voir choix MAP ou TOP) .

Par MMS, les fichiers ne peuvent qu'être lus ou transférés en lecture. Ils peuvent être supprimés ou renommés. Enfin il est possible d'obtenir des informations sur un répertoire distant.

Primitives utilisées : fileOpen
fileClose
fileRead
obtainFile
fileRename
fileDelete
fileDirectory

5.4. ELEMENTS DE PROTOCOLE.

La plupart de ces services sont des services confirmés. Seuls cinq d'entre eux ne le sont pas et sont simplement transmis du client au serveur :

informationReport
unsollicitedStatus
abort
reject

ou du serveur au client :

eventNotification

L'automate correspondant ne comporte alors qu'un seul état et le service ne peut être annulé quand il a été lancé.

Les service confirmés demandent une réponse de la machine serveur distante.

L'automate reste assez simple. Il permet toutefois d'**annuler** une requête (par cancel) qui peut ou non avoir déjà été exécutée; si le service demandé est déjà exécuté l'annulation n'a pas été prise en compte et on reçoit une confirmation négative au "cancel".