

# **Contribution aux Messageries Industrielles**

**E. Gressier-Soudan**

## **Habilitation à Diriger des Recherches**

**LIFL – Lille**

Soutenue le 19 Décembre 2002

Président	:		Professeur à l'Université de Lille 1
Directeur	:	J-M. Geib	Professeur à l'Université de Lille 1
Rapporteurs	:	J-D. Decotignie	Professeur à l' Ecole Polytechnique Fédérale de Lausanne
		G. Pujolle	Professeur à l'Université Paris 6
		M. Riveill	Professeur à l'Université de Nice
Jury	:	J-P. Arnaud	Professeur titulaire de Chaire au Conservatoire National des Arts et Métiers
		G. Florin	Professeur au Conservatoire National des Arts et Métiers
		F. Horn	Ingénieur Kelua
		S. Natkin	Professeur au Conservatoire National des Arts et Métiers

*A Romain, Flora et Léo.*

**Résumé**

**Mots Clefs**

**Abstract**

**Key Words**

## **Remerciements**

## Table des Matières

<b>1 INTRODUCTION.....</b>	<b>7</b>
1.1 ORIGINE DES TRAVAUX DE HDR.....	8
1.2 UN PROJET DE RECHERCHE APPLIQUEE.....	9
1.3 THEMES DE RECHERCHE DU PROJET.....	10
1.4 PLAN DU DOCUMENT.....	14
<b>2 CONSTRUCTION D'UNE MESSAGERIE INDUSTRIELLE ORIENTEE OBJET .....</b>	<b>15</b>
2.1 INTRODUCTION .....	15
2.2 CONCEPTS CLEFS DE MMS.....	16
2.2.1 <i>Le modèle de communication MMS</i> .....	16
2.2.2 <i>Définition des abstractions MMS</i> .....	16
2.2.3 <i>Comparaison MAP/CORBA</i> .....	17
2.3 MOTIVATIONS INITIALES, OBJECTIFS ET APPROCHE SUIVIE.....	17
2.3.1 <i>MMS et les protocoles de l'Internet, besoins utilisateurs</i> .....	18
2.3.2 <i>Modèles pour une Messagerie Industrielle de type MMS sur TCP/IP</i> .....	20
2.4 CONCEPTS CLEFS DE LA NORME CORBA.....	22
2.4.1 <i>Apport du bus à objets CORBA</i> .....	22
2.4.2 <i>Comparaison MMS/CORBA et les avantages de l'approche CORBA</i> .....	23
2.5 CONCEPTION D'UNE MESSAGERIE INDUSTRIELLE ORIENTEE OBJET SUR CORBA : OO-MMS.....	25
2.5.1 <i>Adaptation du modèle de communication</i> .....	25
2.5.2 <i>Implantation des services MMS</i> .....	27
2.5.3 <i>Passage d'une spécification ASN.1 à une spécification IDL</i> .....	28
2.6 PROTOTYPAGES D'UNE MESSAGERIE INDUSTRIELLE ORIENTEE OBJET .....	31
2.6.1 <i>Architecture d'un prototype OO-MMS</i> .....	31
2.6.2 <i>Les Différents Prototypes</i> .....	32
2.6.3 <i>Projet COCA</i> .....	33
2.7 CONCLUSION.....	33
<b>3 VERS UNE MESSAGERIE INDUSTRIELLE ORIENTEE OBJET TEMPS REEL.....</b>	<b>35</b>
3.1 INTRODUCTION .....	35
3.2 CONCEPTION D'UNE MESSAGERIE INDUSTRIELLE ORIENTEE OBJET PORTABLE A QOS SUR ATM.....	36
3.2.1 <i>Approche ORB flexible pour la qualité de service</i> .....	37
3.2.2 <i>L'environnement AJAX</i> .....	40
3.2.3 <i>Conclusion</i> .....	44
3.3 UNE MESSAGERIE INDUSTRIELLE ORIENTEE OBJET AVEC DES CARACTERISTIQUES TEMPS REEL.....	45
3.3.1 <i>Caractéristiques Principales de TASE.2</i> .....	45
3.3.2 <i>OpenTAZ, l'implantation objectifiée de IEC-TASE.2 sur CORBA</i> .....	48
3.3.3 <i>Conclusion</i> .....	51
3.4 PERSPECTIVES ET CONCLUSION .....	58
<b>4 CONCLUSION.....</b>	<b>59</b>
4.1 BILANS.....	59
4.1.1 <i>Bilan Scientifique</i> .....	59
4.1.2 <i>Bilan Quantitatif</i> .....	60
4.2 PROSPECTIVE ET PERSPECTIVES.....	62
4.3 DERNIERES IMPRESSIONS .....	64
<b>5 BIBLIOGRAPHIE.....</b>	<b>66</b>
5.1 REFERENCES EXTERIEURES AU PROJET DE HDR .....	66
5.1.1 <i>Conférences et Ouvrages</i> .....	66
5.1.2 <i>Normes et Standards</i> .....	69
5.1.3 <i>Rapports de Projets Européens, Documentations Techniques et Rapports de Recherche</i> .....	70
5.1.4 <i>Thèses</i> .....	71
5.1.5 <i>Documentations issues du Web</i> .....	71
5.2 BIBLIOGRAPHIE PERSONNELLE RELEVANT DE LA HDR.....	72
5.2.1 <i>Revue Internationales</i> .....	72

5.2.2	<i>Revue Nationale Electronique</i> .....	72
5.2.3	<i>Chapitre de Livre</i> .....	72
5.2.4	<i>Conférences d'Audience Internationale</i> .....	72
5.2.5	<i>Workshops à Audience Internationale</i> .....	73
5.2.6	<i>Conférences à Audience Nationale</i> .....	74
5.2.7	<i>Rapports de Recherche CEDRIC</i> .....	74
5.2.8	<i>Rapports d'Etude sur contrats avec des Industriels</i> .....	75
5.3	TRAVAUX D'ETUDIANTS LIES A LA HDR .....	76
5.3.1	<i>Thèses</i> .....	76
5.3.2	<i>DEA</i> .....	76
5.3.3	<i>Mémoires d'ingénieurs</i> .....	76
5.3.4	<i>Stages de Laboratoire</i> .....	77

## 1 Introduction

Ce document présente les résultats de mes travaux de recherche entrepris à partir d'octobre 1993 jusqu'à ce jour, dès mon retour au laboratoire de recherche CEDRIC du CNAM, après une année de disponibilité pour convenances personnelles dans une entreprise créée en Septembre 1992 par le Professeur S. Natkin: la société CESIR. CESIR avait un objectif double: faire du conseil scientifique dans le domaine des architectures de réseaux pour les applications d'informatique industrielle et participer à l'émergence d'activités artistiques multimédia, et plus particulièrement, d'arts numériques.

Le sujet de mon Habilitation à Diriger des Recherche (HDR), est d'étudier la construction d'une mémoire répartie partagée d'objets avec des caractéristiques temporelles, encore appelée base de données temps réel répartie partagée<sup>1</sup>, à l'aide d'une messagerie industrielle. Cette construction est validée par étape à l'aide de prototypes.

Un service de gestion d'une base de données temps réel répartie partagée est proposé traditionnellement par des environnements de type bus de terrain [CIA98], ou encore [FEL02] pour une vue réactualisée et synthétique de leurs caractéristiques principales. Il existe plusieurs familles de bus de terrain, suivant qu'ils gèrent des équipements proches ou non du procédé physique. Les services qui sont proposés dans mon travail visent des environnements adressés habituellement par des réseaux comme FIP ou PROFIBUS. Ces types de bus servent à relier des organes de calcul qui eux-mêmes gèrent et contrôlent des capteurs et des actionneurs en réseau pour le compte de calculateurs de supervision.

L'architecture des bus de terrain correspond à un réseau en trois couches (1, 2 et 7 dans le modèle ISO-OSI), Mini-MAP [NUS91] en est un exemple type. Ils supportent une messagerie industrielle et permettent de gérer de manière répartie des variables représentant des données de production consultées ou modifiées par le procédé et par la supervision. Notre contribution consiste à proposer une approche nouvelle offrant le même niveau fonctionnel aux applications d'informatique industrielle que les bus de terrain mais fondée sur le concept de système réparti à objets avec des caractéristiques temps réel. Une approche système réparti et orientée objet correspond mieux aux besoins des applications d'informatique industrielle comme nous avons pu nous en rendre compte lors de l'élaboration de besoins utilisateurs [CCE93].

Au moment de la rédaction de ce rapport, l'objectif initial peut être considéré comme quasiment atteint avec le prototype OpenTAZ [OPE02]. Toutefois, nous n'avons pas réussi à tirer profit des résultats issus des différents travaux sur la cohérence des données et le temps

---

<sup>1</sup> Le terme de mémoire répartie partagée de variables est très probablement le plus représentatif, dans le domaine des systèmes répartis, du service qu'offre un bus de terrain aux applications industrielles, mais le terme admis dans cette communauté bien spécifique reste base de données temps réel répartie. Le qualificatif "temps réel" fait référence au fait que les données résident en mémoire d'exécution plutôt que sur disque (bien que ça soit parfois possible dans certaines applications et si la base n'est pas répartie) et obéissent à des contraintes de cohérence temporelle comme la fraîcheur ou la péremption des informations notamment. Dans certaines solutions, on peut accéder à ces bases de données temps réel en utilisant un langage de requête comme SQL.

réel. Force est de constater par ailleurs que la communauté des fournisseurs de bus de terrain fait migrer l'architecture de ses produits avec des objectifs équivalents à ceux établis dans cette HDR : DeviceNet-ControlNet-EtherNet/IP [BRO01] [SCH01], PROFIBUS-PROFINet [POP01], ModBus-IDA [CAM01] ou Foundation Fieldbus avec HSE [FIE02]<sup>2</sup>, mais peut-être que le couplage bus de terrain et plate-forme objet est le mieux accompli à l'aide de la spécification OPC/DX [CAH01] [OPC02] approche que nous n'avons pas eu l'occasion d'évaluer.

Les paragraphes suivants présentent mon parcours de HDR et essaient d'exhiber la cohérence qui relie l'ensemble du travail . Ils donnent l'origine des travaux à travers mon parcours professionnel. Ils décrivent l'ensemble des axes des différents thèmes d'étude, même ceux qui ne sont pas couverts par la HDR et qui représentent pourtant une part significative de ces années passées (fin 1993 à fin1998). Pour mieux souligner cette cohérence, les travaux des différents étudiants qui sont passés sur le projet: thèses DEA, et mémoires d'ingénieur.

## 1.1 Origine des travaux de HDR

L'origine de mes travaux de recherche est multiple. Mes travaux prennent source tout d'abord dans mon expérience industrielle hors de l'univers enseignement-recherche, c'est un aspect dominant:

- Initialement, j'ai été ingénieur logiciel dans l'entreprise SEMA GROUP, j'ai eu l'opportunité de travailler sur une architecture de réseau local industriel à partir du 1<sup>er</sup> Novembre 1985. Mon premier travail concernait le réseau local industriel ARLIC (Architecture de Réseau Local Industriel CERCI) [NAT87], un réseau fondé sur une approche ISO au-dessus d'un Ethernet redondant. A l'époque, cette architecture était révolutionnaire, les versions récentes des bus de terrain proposées par les principaux fournisseurs du marché s'appuient maintenant aussi sur Ethernet dans sa version commutée et font l'objet d'un consortium, AIONA [IAO02]. Une analyse de l'adéquation d'Ethernet aux contraintes des bus de terrain est donnée dans [présentation Decotignie a FET01]. Au préalable, à la fin de ma thèse, dès janvier 1985, je m'étais familiarisé avec l'architecture d'ARLIC à travers une modélisation en Réseaux de Petri Stochastiques du réseaux local pour une application de contrôle-commande dédié à la production d'énergie électrique [GRE87] [NAT86]. J'ai poursuivi par la conception et l'implantation de la fonction supervision d'ARLIC [ref à la V1 à retrouver], à cette époque, la fonction gestion et supervision de réseaux (Network Management) était loin d'être normalisée. On peut noter que le premier mémoire d'ingénieur encadré en thèse associait gestion réseau et intelligence artificielle [TUA87]. Cette première expérience sur un réseau local industriel m'a familiarisé avec les noyaux temps réel, Ethernet, les protocoles de communication OSI et les contraintes de performance des applications d'informatique industrielle distribuées.

---

<sup>2</sup> Foundation Fieldbus intègre l'approche Ethernet TCP-UDP/IP avec HSE (High Speed Ethernet), mais ne semble pas en l'état de nos connaissances avoir intégré une approche objets distribués coopérants de façon aussi avancée que ses concurrents. Ce propos peut être modéré si on considère maintenant HSE avec OPC. Par contre, la définition de blocs fonctionnels spécifiques à des services adaptés aux applications fait penser à une véritable approche composants.

- Mon second poste a été administrateur systèmes et réseaux au CNAM de fin 1987 à fin 1989. J'y ai découvert l'Internet en réseau local, et le système Unix. Cette expérience m'a fait découvrir la communauté Unix, la famille des protocoles TCP/IP de la couche 3 à 7, et le potentiel de l'Internet. Cette expérience m'a aidé pour faire des propositions modernisant l'architecture des applications d'informatique industrielle. Par exemple, dès 1995, nous proposons [LEB95a] [LEF95b] [GRE95] l'utilisation de technologie la Web pour faire de la supervision légère distante, ainsi que l'utilisation de SNMP comme alternative à une messagerie industrielle.
- Enfin, une année passée dans une entreprise en création, CESIR, d'octobre 1992 à août 1993 m'a plongé dans l'univers du conseil en architectures pour les réseaux industriels, notamment lors de la participation à deux projets de la Communauté Européenne : ITCIM [ITC92] [ITC93], et CCE-CNMA [CCE94]. J'y ai découvert les besoins des industriels du domaine (utilisateurs, intégrateurs, fournisseurs). Cette année passée dans le monde industriel, les domaines dans lesquels j'ai dû intervenir et les observations que j'ai pu faire, ont été déterminantes pour l'initiation de mes travaux de HDR.

D'un point de vue académique, mes travaux prennent source aussi dans les activités d'enseignement et de recherche menées au sein de l'équipe Systèmes Répartis du Professeur Gérard Florin. La découverte des Systèmes Répartis date du début de ma thèse en 1984 à travers l'enseignement du, alors jeune projet INRIA, le micro-noyau ChorusOS [ROZ92], en travaux dirigés à l'université Paris 6. Les trois premières années comme Maître de Conférences avec le Professeur G. Florin m'ont permis de vraiment découvrir ce domaine et d'en acquérir les principes fondamentaux.

Le début de mon HDR s'est traduit par l'instanciation d'un sous-groupe dans le thème Systèmes Répartis Ordonnés et Temps Réel prenant le nom de projet SAFFRES<sup>3</sup>, que j'ai complètement coordonné d'Octobre 1993 à Décembre 1998 [GRE02]. L'ensemble de ces travaux se sont poursuivis au sein de l'équipe CEDRIC Multimédia sous la forme d'un groupe "services systèmes pour les applications multimédia répartis", aujourd'hui avec les projets: RIAM DISERTIVI avec TF1, Alcatel et Nexstream, et le concert réparti avec l'IRCAM.

## 1.2 Un Projet de Recherche Appliquée

Mon travail se caractérise par le souci de maintenir un lien, aussi souvent que possible, avec le domaine industriel et applicatif, et en particulier à l'informatique industrielle. Les travaux relatifs à la HDR peuvent être classés comme un projet de recherche à vocation pré-industrielle. La nature de mon travail m'a toujours placé à la frontière du domaine académique et du domaine industriel, les contraintes applicatives et architecturales de l'informatique industrielle nourrissant les réflexions sur la mise en œuvre de concepts issus de la recherche universitaire.

En conséquence, nous avons développé une approche d'intégrateur de technologies et de concepts, nous arrêtant à la preuve de faisabilité. Le corollaire à cette approche est

---

<sup>3</sup> SAFFRES fait référence à un site d'Escalade près de Dijon. A l'époque plusieurs projets en systèmes répartis portaient de noms à des lieux d'escalade : SIRAC à Grenoble (INRIA-ALPES/IMAG), Larchant à Rocquencourt (INRIA).

l'utilisation, autant que possible de produits industriels (par exemple pour la partie système temps réel, ChorusOS de Chorus Systems, racheté par Sun Microsystems, et devenu Jaluna depuis la mi-2002, a été longtemps la base de nos prototypages, plus récemment nous sommes passés à pSOS+ et probablement prochainement à VxWorks de Wind River) et une liaison permanente avec des industriels pour valider mes objectifs et les solutions développées (Aérospatiale, Alcatel Alsthom, France Telecom, EDF R&D notamment). Ce choix m'a aussi semblé nécessaire compte tenu des attentes de formation du public étudiant au CNAM.

Dans la mesure où nous nous sommes intéressés à des domaines applicatifs pour lesquels nous n'avions pas de compétences, nous avons été amenés à coopérer avec des équipes de recherche de ces domaines. Dans le cadre de la messagerie industrielle appliquée à la productique, nous avons travaillé avec l'équipe GRPI de l'IUT de St Denis, plus particulièrement avec R. Boissier [GRE99] [BOI01]. Dans le cadre des travaux sur la cohérence appliqués au calcul parallèle pour la biologie, nous avons travaillé avec le Centre de Calcul Scientifique de l'Institut Pasteur en la personne de C. Letondal (service dirigé par L. Bloch, ancien directeur du centre de ressources informatiques du Cnam) et avec l'unité de recherche sur les rétrovirus en la personne de M. Trütwin-Müller (unité dirigée par le Professeur Barré-Sinoussi, co-découvreur du virus du SIDA) [BEA97] [MUL99].

### 1.3 Thèmes de recherche du projet

L'objectif formel de construire une base de données temps réel répartie partagée d'objets équivalente au service offert par un bus de terrain, à l'aide d'une messagerie industrielle orientée objet n'a pas été clairement affirmé immédiatement dans mon projet mais pendant l'été 1997 avec T. Cornilleau et F. Weis après leur thèse [COR97a] [WEI96]. C'était le résultat d'investigations menées conjointement sur trois thèmes connexes: cohérence logique et temporelle de données réparties partagées, plates-formes temps réel, et messagerie industrielle orientée objet. Cohérence des données et messagerie industrielle sont deux aspects fondamentaux des environnements de communication en milieu industriel tels qu'ils sont décrits dans [DEC01].

Les principales études menées par rapport à ces trois thèmes sont les suivantes :

- **Cohérence des données :** Au début des travaux sur ce thème, nous nous sommes orientés vers l'ingénierie de serveurs de cohérences pour une Mémoire Virtuelle Répartie Partagée sur des machines de type PC. L'objectif premier était d'étudier les services nécessaires à l'élaboration d'une mémoire répartie partagée fournissant un service fonctionnellement équivalent à celui d'une mémoire réfléchie (reflective memory) [VME96] mais en logiciel donc adressant un temps réel moins contraint que des produits comme SCRAMNet (Shared Common RAM Network)[SYS02]. Le thème de la cohérence des données est détaillé ci-dessous car il n'est pas traité dans la présentation de la HDR. Plus précisément, nous avons étudié :

L'influence de la causalité dans la gestion de la cohérence d'une mémoire répartie partagée (MRP). La notion de causalité définie par Lamport en 1978 [LAM78] a été reprise et appliquée au problème de l'accès à une mémoire par différents auteurs. Elle est baptisée cohérence causale [AHA91a], et a bénéficié de nombreuses extensions: travaux théoriques, spécifications d'algorithmes, mises en oeuvre, applications et propositions d'extensions

[AHA91b] [RAY92] [BOY92] [PUA93] [AHA93] [AGR94] [JOH94] [GAL95a] [GAL95b] [RAY95a] [RAY95b] [JOH95] [COR95] [COR96a] [COR96b] [THI96a] [THI96b] [BOR96] [COR97a] [COR97b] [COR97c] [THI97]. L'utilisation de la causalité, bien que d'un usage délicat, offre une alternative intéressante à la cohérence forte la plus connue et la plus facile à appréhender. Il est possible aussi de combiner des cohérences uniformes pour offrir des visions différentes d'un même objet [COR96]. La cohérence causale offre une propriété logique, mais les applications d'informatique industrielle requiert aussi des propriétés temporelles. Nous avons proposé la cohérence causale temporelle [COR97b] qui permet de gérer des variables partagées réparties en se rapprochant du service offert par à un bus de terrain. En conclusion des travaux sur la cohérence des données pour les applications d'informatique industrielle, on peut se préoccuper de contraintes de Qualité de Service relatives à la cohérence comme on parle de Qualité de Service temporelle (gigue, latence, échéances, taux de non respect des échéances) qui est quelque chose de plus classique dans les systèmes répartis à contraintes temps réel.

La réalisation au-dessus du noyau CHORUS d'un service de mémoire répartie partagée générique permettant de contrôler des données avec plusieurs types de cohérences. Ce travail prend sa source dans les résultats de la thèse de M-I. Ortega [ORT92], la première mémoire répartie cohérente implantée sur ChorusOS, et de sa modélisation [COR92]. Nous avons obtenu plusieurs prototypes de protocoles de cohérence [COR95] [BOL97] [THI97] [COR97a]. Ce projet nous a amené à tester les mécanismes les plus avancés proposés dans le micro-noyau : les appels de procédures légers (LAP, Local Access Point) et le proxy-mapper [BEC97]. De façon connexe, un service de persistance a été développé [BON96]. Ce projet a été mené en coopération avec Chorus Systèmes, et plus spécifiquement M-I. Ortega. Cette série de travaux s'est terminée par le prototypage d'un service de migration d'objets construit au-dessus de serveurs de cohérence. Nous avons vu une ébauche de débouché pour ce type d'architecture avec les PABX Alcatel [ROS96] mais il n'y a pas eu de suite. La conclusion de cette voie d'étude tient au fait que le mécanisme de mémoire répartie partagée reposant sur un grain de gestion de niveau page est inapproprié et même contre performant pour faire de la cohérence d'objets [MOL99].

Etudier l'utilisation de protocoles de cohérence faibles et hybrides pour des applications. J'ai pris contact avec le professeur Willy Zwaenepoel de l'Université de Rice au Texas, invité à la conférence MPR'96 à Bordeaux dont j'ai été l'un des co-organisateurs. Il est le fondateur du projet TreadMarks [AMZ96] qui a produit la bibliothèque du même nom qui permet la programmation d'applications parallèles au-dessus d'une mémoire répartie partagée. TreadMarks implante une cohérence relâchée paresseuse, "lazy release consistency". Dans le cadre d'un projet en Biologie avec le service d'Informatique Scientifique de l'Institut Pasteur, nous avons expérimenté l'adéquation de l'environnement TreadMarks pour exécuter un algorithme de recherche phylogénétique pour le classement de séquences d'ADN issues de virus du SIDA [BEA97], [MUL99]<sup>4</sup>. Le résultat obtenu permet de diviser d'un tiers le temps de calcul (gain d'une semaine). La suite de ces travaux a visé à expérimenter TreadMarks sur le noyau ChorusOS sous la forme d'un ensemble de serveurs de cohérence [BOU97] ou sous la forme de bibliothèques [SAU99]. Nous n'avons pas continué à explorer cette voie car elle nous menait à s'investir dans le domaine du calcul parallèle, ce qui m'éloignait de mes objectifs initiaux.

Il y a eu des travaux plus "exotiques" par rapport à l'axe principal de recherche. Par exemple, l'association de propriétés d'ordres sur groupes de diffusion pour mettre en œuvre des protocoles de cohérence [CAL95a] [CAL95b] [CAL96] ou l'évaluation de mécanismes de points de reprise associés à une mémoire répartie partagée à cohérence relâchée paresseuse

---

<sup>4</sup> Cette référence bibliographique est du domaine de la biologie.

[BAI98] ont été des voies qui n'ont jamais été approfondies au delà des stages de DEA ou des mémoires d'ingénieur qui en ont fait l'objet.

- **Plate-forme temps réel** : Le prototypage et l'intégration de technologies système et réseau à destination du temps réel réparti qui est un thème sous-jacent aux deux autres. De premiers travaux ont préparé le terrain à travers une expérience sur l'ordonnement par échéance avec le noyau ChorusOS [GAU94]. Le stage de DEA de C. Lizzi [LIZ95] sur l'intégration d'un coupleur FDDI à une version modifiée de Chorus pour du multimédia distribué puis le mémoire d'ingénieur de M. Arazo [ARA96] sur un protocole de transport temps réel, tous deux au CNET à Issy-les-Moulineaux dans l'équipe de J-B. Stéfani, correspondent à cet objectif. L'implantation d'un modèle synchrone fondé sur Esterel [BER87] réparti par couplage à Chorus/COOL dans le cadre d'un stage de F. Singhoff chez Dassault Aviation [SIN96] participe à cette problématique de recherche d'une plate-forme temps réel distribuée.
- La thèse de C. Lizzi pousse la démarche avec ChorusOS le plus loin qu'il nous a été possible, elle explore la mise en œuvre d'une architecture à QoS du micro-noyau jusqu'au bus logiciel en intégrant tous les éléments de la plate-forme autour d'un réseau ATM [LIZ99c]. Ce projet mené dans le cadre de l'entreprise CSTI (Compagnie des Signaux Technologie Informatique) avait des objectifs communs au projet MidART du MERL (Mitsubishi Electric Research Laboratory) [GON97]. La plate-forme de C. Lizzi, bâtie à partir d'un ChorusOS étendu avec des propriétés temps réel telles qu'un ordonnancement par échéance, l'héritage de priorités de bout en bout à travers un réseau ATM, une gestion mémoire en zéro-copie, complétée avec le proxy-mapper d'E. Becquet [BEC97] implantant une cohérence causale temporelle [COR97b] aurait probablement fait émerger une architecture équivalente à RT-CRM [SHE00], voire proche de CreMeS [CHU00] avec nos travaux sur la messagerie industrielle au-dessus de CORBA, MMS-CORBA [SIM96] [GUY97a] [GUY97b] [GUY02].
- Les conséquences des travaux de cette thèse ont permis d'amener des résultats architecturaux [LIZ00] qui nous ont aidés dans le choix de composants techniques sur étagère [BAC00] [GRE00b] [GRE01b] pour l'informatique industrielle d'EDF dont les besoins ont été préalablement étudiés [BAC98a] [BAC98b] [BAC99a] [BAC99b] [BAC99c] [BAC99d]. Pour étayer ces travaux, une évaluation fonctionnelle de noyaux temps réel du commerce avec un benchmark a aussi été menée [BAC98c] [BAC98d] [VxWorks] [pSOS+] [LynxOS] [DEC99] [BEC01a]. Une autre étude de noyaux pour l'informatique industrielle avait d'ailleurs été menée auparavant pour Alcatel Alsthom Recherche [CHAB97a] pour sa nouvelle génération de superviseur, le P3200NG [CHA97b] [CHA97c] [CHA98], celle-ci ne comportait pas de benchmark. L'étude pour EDF marque d'ailleurs un tournant vis à vis du choix de notre plate-forme de référence ChorusOS, devenue un produit Sun Microsystems. Cette étude s'est conclue par l'étude de bus logiciels pour l'informatique industrielle EDF CORBA [BEC00], DCOM [LOG00], et de l'environnement Java [REV00] et par la proposition et le prototypage d'une architecture de processeur pour la supervision distante de petite et moyennes unités de production d'énergie [REV01b] fondée sur le noyau pSOS+, une machine virtuelle Java temps réel, PERC [NIL96] [NEW00], des

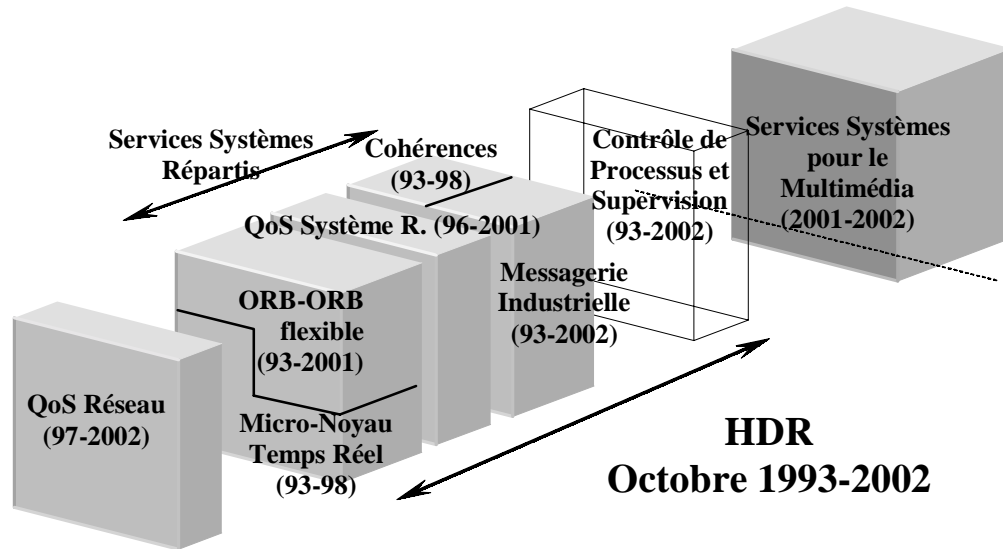
servlets Java [REV01a], une IHM accessible via un navigateur web [GUI00]. Ce travail est représentatif d'une certaine tendance visant l'intégration de l'approche Java et de la technologie Web dans les équipements d'informatique industrielle, éventuellement très proches du procédé, on peut citer à titre d'exemple la solution de commande à distance d'automate programmable (PLC, Programmable Logic Controller) par TRiLOGI [TRI02]. Notons qu'on peut parfois s'étonner de l'enthousiasme de certains industriels à abandonner leurs valeurs de références pour vénérer ces nouvelles idoles, on peut lire par exemple "*...group is proposing standards for real time distributed automation based on Ethernet and Web technologies*" dans [REE02], rappelons que le thème Ethernet et temps réel fait actuellement l'objet d'un débat de fond dans la communauté bus de terrain [DEC01a], que TCP est reconnu comme mal adapté au trafic à contraintes temps réel (effets du contrôle d'erreur, du contrôle de flux et du contrôle de congestion notamment) et que UDP est préférable, et enfin que l'utilisation du format XML, très verbeux, avec http invite à la réserve pour l'acheminement des données temps réel.

- **Messagerie Industrielle** : La description ci-dessous est plus succincte car elle fait l'essentiel du présent document.

Nous avons tout d'abord voulu étudier l'intégration des protocoles TCP/IP à la messagerie industrielle MMS (Manufacturing Message Specification) de l'ISO [VAL92], désignée par ISO-MMS dans la suite de ce document. Après un premier prototypage mettant en oeuvre la norme ISO-MMS au-dessus des RPC de Sun Microsystems [LEF95a] [LEF95b] [GRE95] et même une projection avec l'IPC de ChorusOS [GRE96], nous avons abouti à un prototype transposant un sous-ensemble de MMS à l'environnement OMG/CORBA offert par COOL/CHORUS [GUY97a] [GUY97b] [GUY02]. Deux évolutions se sont produites : rendre les objets de la messagerie industrielle migrables par une ré-écriture du prototype OO-MMS en Java [LAU99] [GRE99b], et prendre en compte des contraintes temps réel sur les interactions entre clients et objets serveurs [BOI98] [BOI01]. Ce dernier aspect nous a éloignés de l'approche CORBA pour choisir un modèle d'interaction plus adapté : RM-ODP et son extension ReTINA [BLA97]. En conséquence, nous avons adopté pour nos prototypages une plate-forme plus conforme à ce modèle : l'ORB flexible Jonathan. Parallèlement, il nous a semblé important de pousser le prototypage jusqu'au pilotage d'une commande numérique [EPI98]. Cette expérience a été menée en coopération avec le groupe de recherche en productique GRPI de l'IUT de St Denis et plus particulièrement Raymond Boissier qui a été en charge de la partie métier.

L'ajout de composants capables de supporter des interactions avec des contraintes temps réel n'apporte pas nécessairement des propriétés temps réel à une messagerie industrielle. Pour cela, nous nous sommes inspirés de la norme IEC TASE.2 [IEC96a] [IEC96b] [KEM96] [IEC97] [EPR98a] bâtie au-dessus de ISO-MMS. La messagerie industrielle a été du coup simplifiée et permet d'être beaucoup plus proche d'un service de gestion de variables réparties comme dans les bus de terrain. Ce travail fait suite à l'étude de besoins utilisateurs provenant d'EDF R&D dans le domaine du contrôle de production d'énergie électrique.

Pour résumer les thèmes abordés, mon travail de HDR est à l'intersection des disciplines réseaux, systèmes répartis et systèmes répartis temps réel pour l'informatique industrielle. Le schéma ci-après donne une vue synthétique, sous la forme de volumes, des sujets explorés par les différents travaux associés à la HDR.



*Figure 1. Domaines de recherche explorés pendant la HDR*

#### 1.4 Plan du document

Ce document est organisé de la façon suivante. Le Chapitre 2 résume les aspects clefs de la conception d'une messagerie industrielle orientée objet à partir de la norme ISO-OSI MMS. Le Chapitre 3 décrit les étapes menant à une extension temps réel, et l'état d'avancement de cette extension. Le Chapitre 4 conclut le document en présentant le bilan et les perspectives de mon travail.

## 2 Construction d'une Messagerie Industrielle Orientée Objet

### 2.1 Introduction

Ce chapitre décrit la conception et le prototypage d'une messagerie industrielle orientée objet à partir d'une approche bus logiciel de type CORBA[SIEG96]<sup>5</sup>. Ce travail prend comme modèle de messagerie industrielle la norme MMS [NUS91] [VAL92] [ISO90] [ISO91a] de l'ISO. L'objectif final de ce travail, qui ne sera présenté qu'au chapitre 3, est d'offrir une messagerie industrielle capable de prendre en compte des contraintes temps réel réparties portant sur les interactions entre objets applicatifs comme par exemple : temps d'exécution d'invocation borné, échéance associée à une invocation, gigue d'invocation... A cette fin, la partie communication utilisera la capacité d'ATM à gérer la Qualité de Service temporelle (QoS, *Quality of Service*).

Brièvement, ce paragraphe donne un aperçu de l'ensemble du travail réalisé autour de ISO-MMS. Cette messagerie a tout d'abord été développée au-dessus d'un bus logiciel conforme à la norme CORBA en C++ sur le micro-noyau temps réel ChorusOS [CHO96a], [CHO96b]. Les versions ultérieures ont été réécrites en Java, elles peuvent s'exécuter sur différents bus logiciels et sur différents systèmes d'exploitation classiques de type Windows 95/98/NT ou Unix. L'utilisation de Java a permis de rendre notre messagerie orientée objet portable, c'est à dire indépendante du système d'exploitation, et du matériel. Pour permettre la prise en compte des paramètres de QoS et bénéficier du support ATM dans les interactions de la messagerie, nous avons utilisé un bus logiciel modulaire et extensible, Jonathan, fourni par le CNET [DUM98] [OBJ01].

L'ensemble de ce travail a été présenté à l'OMG, devant le groupe de travail Manufacturing Domain Task Force à Burlingame en Septembre 2000 [GRE00a]. La même proposition a été ébauchée par l'EPFL [MER98] mais plus tardivement. Les travaux publiés à ECOOP'97 ont été repris, une autre équipe a étendu la proposition d'architecture en y ajoutant la gestion d'événements MMS [ARI98], on peut indiquer aussi l'implantation d'un équipement virtuel dans ce canevas [ARI01].

Le Chapitre est organisé de la manière suivante. Le paragraphe 2.2. décrit les concepts clefs de ISO-MMS. Le paragraphe 2.3. décrit les motivations qui ont présidé au démarrage de ce projet, nos objectifs, et l'approche suivie. Le paragraphe 2.4. décrit les aspects clefs du standard CORBA qui ont permis l'émergence d'une messagerie industrielle orientée objet. Le paragraphe 2.5. présente la conception de cette messagerie. Le paragraphe 2.6 décrit les prototypes qui en ont découlé. Le paragraphe 2.7 conclut.

---

<sup>5</sup> La référence à CORBA suivant le livre de D. SIEGEL [SIE96] est volontairement ancienne, comme l'architecture de CORBA qui a servi à ce projet, qui était à l'époque CORBA 2.0. Depuis de nombreuses versions sont apparues, aujourd'hui, CORBA 3.0 a vu le jour [OMG02a]. En l'état de la normalisation, aujourd'hui, Minimum CORBA [OMG02b] correspondrait à nos besoins.

## 2.2 Concepts Clefs de MMS

La spécification MAP est née du besoin de connecter d'une manière simple et efficace les différents équipements utilisés dans les applications de productique (automates programmables, commande numérique, robot...). L'architecture MAP peut se résumer à une pile de protocoles OSI sur 7 couches, avec un choix de profils adaptés à un milieu industriel. La norme de messagerie industrielle MMS spécifie le protocole de la couche application mis en oeuvre dans le cadre de MAP. Elle structure le dialogue entre les différentes machines d'un atelier de fabrication, en offrant un ensemble normalisé de services de communication entre équipements de production hétérogènes, ce qui facilite l'interopérabilité. Des normes d'accompagnement ont été définies pour les principaux types de serveurs utilisés : automates programmables, commandes numériques, robots. D'autres normes d'accompagnement visent des services spécifiques à un métier, TASE.2 de l'IEC [IEC96a] [IEC96b] [KEM96] [IEC97] [EPR98a] vise les services de production d'énergie, d'eau ou de gaz.

### 2.2.1 Le modèle de communication MMS

L'architecture MAP met en relation des entités d'application de façon symétrique. Cependant, les échanges entre deux applications utilisant les services MMS sont organisés d'une façon dissymétrique, avec une relation de type client/serveur. Le modèle MMS définit un ensemble de services qui permettent à une application client de piloter et de contrôler à distance un ou plusieurs équipements serveur.

La plupart des services MMS sont de type confirmé. Dans ce cas, l'exécution du service entre le client et le serveur est gérée à l'aide de quatre primitives : requête, indication, réponse positive ou négative, confirmation. La confirmation matérialise l'exécution du service. Le serveur reçoit une primitive d'indication et émet une primitive de réponse, positive ou négative, selon que la demande a pu être satisfaite ou non. MMS offre également quelques services de type non confirmé permettant à un serveur d'envoyer des données en direction d'un client.

Ce modèle client/serveur a la capacité de gérer les envois et les réceptions de primitive de manière asynchrone : une nouvelle demande de service peut être effectuée par un client avant d'avoir reçu la primitive de confirmation correspondant à la précédente. Plusieurs demandes de service peuvent être donc concurremment en cours d'exécution, chacune d'entre-elles étant caractérisée par un identificateur de lancement.

### 2.2.2 Définition des abstractions MMS

Un des aspects novateurs de la norme MMS se situe au niveau du serveur. Ce dernier est composé d'un ensemble d'abstractions dits "objets MMS". Chacune de ces abstractions représente une ressource de l'équipement réel auquel le serveur est lié et peut être manipulé à l'aide des services MMS. L'ensemble des "objets MMS" d'un serveur est regroupé au sein d'une ou plusieurs entités appelées VMD (Virtual Manufacturing Device). Un VMD est donc la représentation abstraite d'un ensemble de ressources d'un équipement réel de fabrication.

La spécification MMS a doté le VMD d'une grande richesse fonctionnelle : de nombreuses abstractions sont définies dans la norme, censées couvrir la plupart des besoins des

applications de productique. Mais cette variété s'accompagne d'une grande complexité de mise en oeuvre. On constate ainsi que seul trois "objets MMS" sont couramment utilisés pour implanter un VMD : la variable, le domaine qui peut contenir à la fois des données (par exemple le code exécutable d'un programme) et des variables, l'invocation de programme qui peut être rattachée à un programme de production s'exécutant au niveau du serveur. Nous considérons ces trois abstractions dans le cadre de l'étude d'une implantation d'un VMD sur CORBA. La figure 2 présente un exemple de VMD.

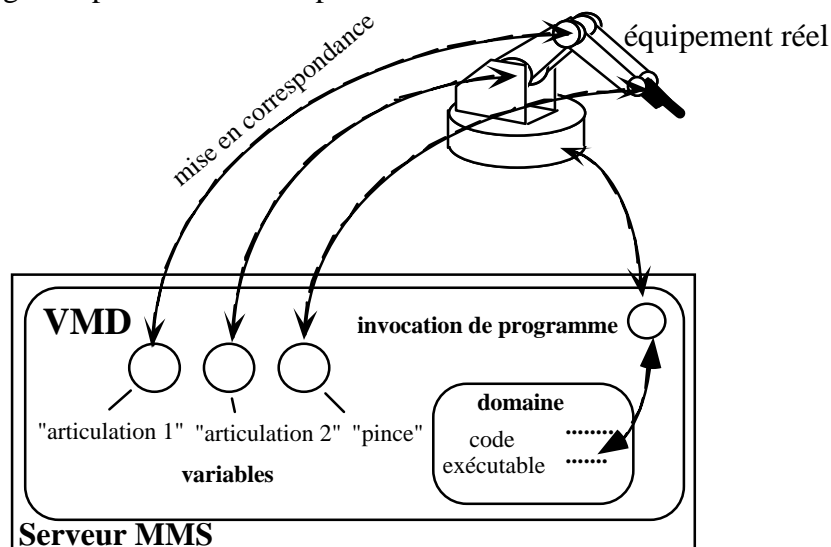


Figure 2 : Lien entre un VMD et l'équipement réel

### 2.2.3 Comparaison MAP/CORBA

Dans le cadre de notre approche, nous avons mis en évidence certaines analogies entre l'architecture MAP et l'architecture CORBA. Tout d'abord, on peut comparer les services de communication offerts par la pile MAP à ceux d'un ORB. Tout comme le fait l'ORB, la pile MAP garantit l'interopérabilité entre des équipements hétérogènes. Néanmoins, MAP repose uniquement sur des services traitant les communications entre un client et un serveur MMS et n'offre pas, contrairement à CORBA, des fonctions étendues pour gérer la désignation et le contrôle des actions concurrentes.

Les services MMS implantés dans un serveur constituent pour un client l'interface d'accès à un VMD. Ces services sont définis dans le langage ASN.1 associé à la couche présentation de la spécification MAP. Ce langage, utilisé par l'ensemble des protocoles d'application définis par l'ISO, permet de décrire les structures de données échangées entre le client et le serveur MMS. De ce point de vue, l'ASN.1 peut être considéré comme un IDL associé à l'environnement ISO. Ainsi, dans notre approche, les services MMS décrits en ASN.1 sont transformés en des spécifications en IDL CORBA. On préserve ainsi les spécifications de la norme ISO-MMS. L'avantage est de bénéficier d'un nouvel environnement d'exécution réparti ajoutant des services de transparence d'accès et de localisation qui n'existent pas dans ISO-MMS.

### 2.3 Motivations initiales, Objectifs et Approche suivie

Notre projet initialement avait un objectif industriel. Nous avons cherché comment tenir compte de l'apport des protocoles de l'Internet pour la conception d'une messagerie industrielle, son objectif a évolué graduellement. Notre premier objectif a donc été de porter la messagerie MMS dans un environnement offrant une bonne intégration des interactions client-serveur au-dessus de TCP/IP. Notre second objectif a tenté d'inclure deux approches nouvelles : l'objet et la répartition. Notre objectif final vise la prise en compte de contraintes temps réel entre objets applicatifs coopérants à l'aide la messagerie industrielle, mais ce dernier aspect est développé dans le chapitre suivant .

### **2.3.1 MMS et les protocoles de l'Internet, besoins utilisateurs**

La pile de protocoles MAP (Manufacturing Automation Protocol) [VAL92] définit un profil normalisé pour le pilotage d'équipements de production hétérogènes en réseau. MAP a été une réponse au besoin de communication grandissant dans le domaine de la productique. La messagerie industrielle MMS (Manufacturing Message Specification) [ISO90][ISO91a], protocole d'application mis en oeuvre dans MAP, couvre les fonctions les plus importantes des environnements de production.

Suite à différentes études des besoins d'utilisateurs, d'intégrateurs et de fournisseurs du domaine de l'informatique industrielle [ITC93][CHE93], nous avons pu constater que les fonctionnalités offertes par le protocole ISO MMS suscitait un vif intérêt de la part d'industriels dans le domaine de la production manufacturière et du contrôle-commande. En effet, MAP et MMS n'ont pas rencontré le succès escompté pour les raisons suivantes : complexité de mise en oeuvre, performances jugées insuffisantes, absence d'interface de programmation normalisée, coût élevé de la pile de communication OSI, promotion insuffisante des produits existants par les fournisseurs d'équipements. Ces différents acteurs auraient préféré utiliser les protocoles de la pile Internet, ce que nous avons tenté de développer [CAD93].

Autre avantage d'une approche de communication utilisant Internet, sa mise en oeuvre permet de bénéficier d'outils supplémentaires : NTP (Network Time Protocol) pour la synchronisation d'horloge, SNMP (Simple Network Management Protocol) pour l'administration réseau, NFS (Network File System) pour le partage de fichiers, FTP (File Transfert Protocol) pour les transferts de fichiers, TFTP (Trivial File Transfert Protocol) pour le téléchargement, HTTP (Hyper Text Transfert Protocol) et la navigation WEB pour de la supervision hyper-média à distance comme présenté dans la figure 3. [GRE95] [RAZ96] [GUI00]... Cet environnement est répandu, éprouvé, bien maîtrisé en terme de connaissances et de savoir-faire. Il contribue donc à diminuer les problèmes d'interopérabilité lors de la phase d'intégration sur site des applications de productique. Ce dernier aspect est non négligeable pour les intégrateurs comme pour les utilisateurs.

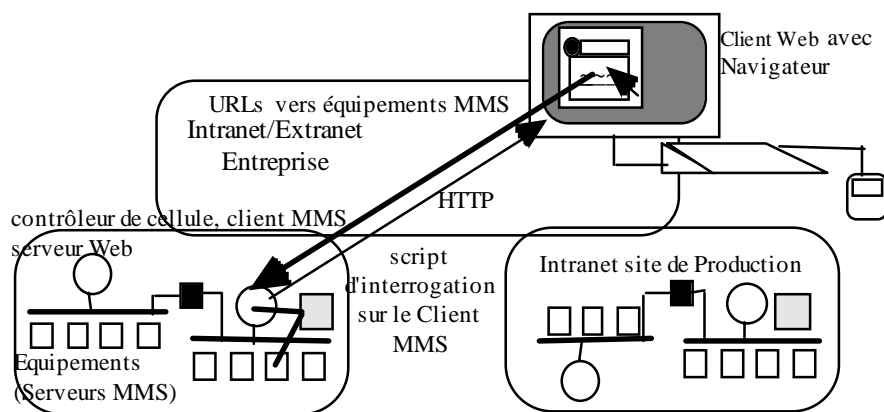


Figure 3 . Supervision en utilisant le Web avec un Navigateur

Les industriels accordent beaucoup d'intérêt à la gestion réseau. La production peut avoir à souffrir des dysfonctionnements du réseau. Cette préoccupation légitime trouve une réponse dans l'offre SNMP des fournisseurs de matériels réseau. Un grand nombre de leurs produits intègrent un agent SNMP. Par ailleurs, le réseau et l'ensemble des contrôleurs de communication peuvent être vus comme une ressource particulière de l'environnement de production [CHE93]. La supervision du réseau est alors capable de compléter la gestion de production en effectuant un suivi de l'activité réseau : détection des incidents, gestion de configuration, surveillance du trafic et des performances. Remarquons par ailleurs qu'il y a certaines ressemblances entre SNMP et MMS. Chacun d'eux offre l'abstraction d'une machine virtuelle qui permet l'accès à des informations concernant un équipement réel. Dans le cas de SNMP, on accède à un contrôleur de communication. Dans le cas de MMS, on accède à équipement de fabrication : le spectre des fonctions couvertes est plus large, il est donc plus difficile de construire l'équivalent de la MIB (Management Information Base). On pourrait imaginer qu'un équipement de production qui n'effectuerait que des opérations d'écriture ou de lecture de variables puisse être piloté en étendant SNMP [GRE95]. Une telle solution ne nécessiterait aucune modification du protocole. L'adjonction de fonctions d'accès à l'équipement dans l'environnement SNMP et l'enrichissement de la base d'informations de gestion réseau (MIB, *Management Information Base*) suffiraient dans ce cas. C'est une solution qui permet le pilotage d'équipements de production à moindre coût et qui pourrait réutiliser le principe de RMON [WAL97].

L'utilisation d'Internet apporte aussi des inconvénients liés à la sécurité des matériels en réseau. Elle nécessite des mécanismes de cryptage et des protocoles de sécurité pour offrir : de la confidentialité (pour ordres de production, télémaintenance par exemple), de l'authentification (pour opérateurs et autres intervenants distants), de l'intégrité (à propos d'acquisition de rapports d'états), et aussi de la non-répudiation (pour certaines commandes modifiant le processus de production comme l'arrêt ou le démarrage). Ce type de besoin a été exprimé par les utilisateurs depuis longtemps, en particulier dans [CCE94]. Il est toujours actuel, mais l'offre des outils s'est grandement complétée, la sécurisation d'un réseau peut s'appuyer aussi sur des filtres et des portes coupe-feu.

### 2.3.2 Modèles pour une Messagerie Industrielle de type MMS sur TCP/IP

L'idée première de ce travail provient directement de l'expérience industrielle que j'ai vécu dans la période 1992-93. Envisager ISO-MMS sur TCP/IP avait commencé à être envisagé au début des années 1990 mais sans remettre en question la hiérarchie OSI, de façon identique aux travaux de M. T. ROSE [ROS90] qui s'appuyait sur le RFC1006[ROS87] pour soutenir la partie haute de la pile OSI. Les quelques industriels qui avaient pris ce risque travaillaient conformément au modèle (b) présenté ci-dessous. Faire des protocoles de communication pour l'informatique industrielle au-dessus de TCP/IP est devenu complètement standard vers la fin des années 1990, environ à partir de 1997-98.

Les protocoles de l'Internet rencontrant un succès grandissant dans le domaine de l'informatique de production, il a donc été naturel d'envisager, comme premier objectif, la prise en compte de ces protocoles dans les nouvelles architectures de systèmes de productique. L'originalité de la pile MAP réside dans sa couche application : la norme MMS. La structure en couche du modèle OSI donne à chaque couche une certaine autonomie. Nous avons donc proposé d'en extraire le protocole MMS pour le mettre en oeuvre dans un nouveau type d'environnement. Plusieurs modèles ont été proposés dans [LEF95a] [LEF95b] [GRES96] que nous rappelons dans la figure 4.

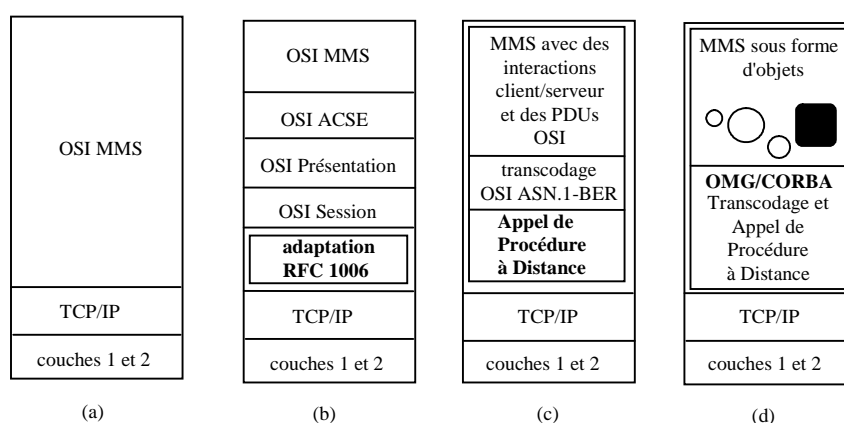


Figure 4. Modèles pour MMS au-dessus de TCP/IP

#### 2.3.2.1 MMS directement au-dessus de TCP/IP

Le modèle (a) semble être l'architecture la plus simple. Elle correspond à mini-MAP [NUS91]. C'est en fait celle qui demande le plus de travail. Les types de données véhiculés par MMS sont potentiellement nombreux et peuvent être échangés entre processeurs avec des représentations internes hétérogènes. Les fonctions de présentation sont donc inévitables. Il faut alors les placer dans l'architecture, autant porter mini-MAP sur TCP/IP.

Nous n'avons pas retenu cette architecture. Elle est toutefois très moderne d'un point de vue industriel aujourd'hui. Elle représente l'architecture type des technologies bus de terrains du niveau contrôle qui effectue une migration vers Ethernet commuté, et TCP-UDP/IP [IAO02]. Cette architecture a fait par ailleurs l'objet de travaux récents [MYO02].

### 2.3.2.2 MMS au-dessus du RFC1006

Dans le modèle (b) de la Figure 4. , on peut porter le protocole de session ISO et l'ensemble des protocoles des couches hautes au-dessus de TCP/IP. Cette solution permet de préserver l'investissement important réalisé dans la réalisation des protocoles ISO des couches hautes. Elle déborde le cadre de MMS, puisque, une fois l'interface session réalisée, tous les protocoles de niveau supérieur peuvent être ainsi portés.

L'adaptation consiste à adjoindre au-dessus de TCP une couche qui réalise un service de Transport OSI Classe 0. Cette couche intermédiaire est décrite dans le RFC 1006 (Request For Comment) [ROS87] mis à jour par le RFC 2126 [POU97]. On peut rappeler que le projet ISODE [ROS90] a adopté cette solution pour expérimenter les couches hautes du modèle OSI dans l'environnement Internet : session, présentation, ACSE, RTSE, ROSE, X500 entre autres. Cette solution a été retenue par Semagroup pour offrir MMS au-dessus de TCP/IP [SEM94]. Suivant le même principe, il est possible de porter MMS au-dessus d'autres familles de protocoles : réseaux propriétaires ou réseaux de PC [NIN92].

### 2.3.2.3 MMS au-dessus de l'appel de procédure à distance

Le mécanisme d'appel de procédure à distance (RPC, *Remote Procedure Call*) présente une analogie avec le modèle d'exécution client/serveur de la messagerie industrielle. Une requête de service effectuée par un client MMS peut être vue comme l'invocation d'une procédure distante d'un serveur. De même, l'exécution d'un service au niveau d'un serveur MMS peut être comparée au déroulement d'une procédure sur un site distant. Toutefois, la correspondance n'est pas absolue puisque dans le modèle client/serveur de MMS, l'envoi d'une nouvelle demande de service n'est pas assujéti à la réception de la réponse de la requête précédente. Le mécanisme de RPC masque l'utilisation de l'interface de transport et les mécanismes de communication sous-jacents. Il propose aussi un nouveau modèle de programmation aux concepteurs d'application de productique. Au lieu d'utiliser l'échange de messages, ils peuvent structurer les interactions entre entités sous une forme procédurale quelle que soit leur répartition.

La norme MMS représente un travail important dans la spécification des échanges de données entre clients et serveurs de tous types, la définition du modèle client/serveur avec le concept de VMD, la structure des objets. C'est le fruit de la réflexion de nombreux experts, il nous a semblé important de conserver ces résultats. Cette hypothèse nous permet de rester proche de la norme. Elle facilite aussi la réalisation ultérieure de passerelles entre les univers OSI et Internet. Ce constat donne une grande importance au format des unités de données échangées formulées en syntaxe ASN.1 dans la norme. Ces formats ont été intégralement repris dans le prototype pour les services de la norme qui ont été implantés [LEF95a] [LEF95b].

Notre premier objectif a été atteint avec la réalisation d'un premier prototype qui utilise l'RPC de Sun Microsystems ONC-RPC [COR90]. Le codage/décodage des unités de données s'effectue suivant les règles ASN.1-BER. Les procédures d'encodage et de décodage ont été générées à l'aide du compilateur ASN.1-C Snacc[SAM93]. Snacc est un logiciel domaine

public qui permet de générer des modules de transcodage en C ou C++ à partir de structures de données décrites en ASN.1.

Le modèle (c) pré-figure les messageries industrielles construites à partir d'XML-RPC, SOAP ou de .NET, comme cela semble aussi le cas dans la proposition OPC/DX [CAH01][OPC02].

Le modèle (d) est une extension à l'objet et au monde CORBA du modèle (c). En effet, on peut voir une invocation de méthode comme l'intégration d'un appel de procédure à distance dans un univers objet.

#### **2.3.2.4 Vers une messagerie industrielle orientée objet**

Notre deuxième objectif dans le projet a été d'intégrer deux approches supplémentaires : l'approche objet et l'approche système réparti. Nous avons donc été amené à travailler sur la solution (d), fondée sur CORBA (Common Object Request Broker Architecture) V2.0 [SIE96], pour implanter MMS dans un environnement objet réparti intégrant TCP/IP. Ce choix a répondu à plusieurs besoins : l'utilisation d'un standard, un environnement permettant la communication entre machines hétérogènes, aussi bien du point de vue système que du point de vue matériel, le support du paradigme objet, un environnement d'exécution réparti et l'interopérabilité entre ORB avec IIOP (Internet Inter-ORB Protocol). Par ailleurs, les abstractions offertes par MMS (appelées objets MMS dans la norme) respectent en partie certains concepts clefs de l'approche objet, en particulier l'encapsulation [WEI96]. Ceci permet de définir les abstractions de base de la norme MMS comme des objets supportés par un environnement CORBA en conservant leur syntaxe et leur sémantique. Cette démarche offre une approche uniforme qui facilite l'intégration des applications de production : l'objet peut alors être le support des méthodes de génie logiciel, des langages de programmation et de l'environnement d'exécution [CCE 94]. Ce deuxième objectif a été atteint par la réalisation d'un prototype OO-MMS [SIM96] [GUY97a] [GUY97b] [GUY02]. Il a été utilisé pour valider la faisabilité d'une architecture de Base de Donnée Temps Réel fondée sur CORBA et programmée suivant une approche synchrone fondée sur LUSTRE [CHA98]. OO-MMS a ensuite été ré-écrit en Java pour fournir une messagerie industrielle orientée objet portable utilisée pour la gestion à distance d'une machine à commande numérique [BOI98] [GRE99a] [LAU99].

### **2.4 Concepts clefs de la norme CORBA**

Le standard CORBA de l'OMG (Object Management Group) fournit un support de développement et de programmation pour la conception d'applications distribuées orientées objet suivant le modèle client/serveur. Dans le cadre de cette étude, nous avons utilisé une implantation de la norme CORBA 2.0. D'autres versions sont actuellement finalisées : de 2.1 à 2.6, ou encore en cours d'élaboration: on attend la 3.0.

#### **2.4.1 Apport du bus à objets CORBA**

La pile OSI peut être perçue comme une architecture de type bus logiciel orienté message. Le passage au bus logiciel de l'OMG semble alors plus naturel. L'OMG a défini une architecture générique qui comprend quatre composants [SIG96]:

- L'ORB (*Object Request Broker*), également appelé bus logiciel, offre une infrastructure de communication (*middleware*). C'est le noyau de l'architecture. Il est chargé de masquer l'hétérogénéité des systèmes et des matériels. Il garantit de plus l'interopérabilité entre les différentes machines qui supportent les objets serveurs.

- Les services communs ou COS (*Common Object Services*) fournissent des fonctions systèmes qui étendent les services rendus par l'ORB: nommage des objets, événements, sécurité, contrôle de concurrence, persistance ... Ces services ont des interfaces spécifiés en langage IDL (*Interface Description Language*) dont le rôle est précisé dans la suite de ce chapitre. Ces services permettent de construire des applications indépendantes des mécanismes sous-jacents du système d'exploitation utilisé. Si on les compare à l'approche de normalisation OSI, ils correspondraient à certains protocoles de la couche application. Ainsi, le service de nommage offre un système de désignation pour retrouver un objet à partir de son nom. Le rôle d'un tel service est très proche de celui de l'annuaire X.500. Parmi les autres COS existants, on peut également citer le service de concurrence d'accès. Ce dernier permet de contrôler l'accès à des objets CORBA partagés, il pourrait remplacer le service de sémaphores MMS. Le service d'événements permet l'abonnement en mode interrogation ou en mode notification à un producteur d'événements, il pourrait être utilisé en remplacement des événements MMS. Le service de persistance permet l'archivage d'objets CORBA, et pourrait servir à remplacer le chargement et le déchargement de domaines MMS.

- Les services applicatifs correspondent à des services déjà développés, soit pour l'aide à la construction et à la gestion d'objets applicatifs, soit par rapport à un domaine d'activité économique. Ainsi, MMS au dessus de CORBA pourrait devenir un service applicatif pour le contrôle/commande ou la productique s'il était normalisé par l'OMG. Nos travaux ont d'ailleurs été présenté à l'OMG dans le domaine de l'informatique industrielle [GRE00].

- Les objets applicatifs ou *Application Objects* sont développés spécifiquement pour l'utilisateur, et donc non standardisés. Ainsi, dans nos prototypes, les objets MMS implantés au dessus de CORBA sont des objets applicatifs.

L'ORB est le support de communication qui permet à un client d'invoquer une méthode associée à un objet désigné explicitement par un nom symbolique, sans référence à sa localisation. Comme pour l'appel de procédure à distance, la requête correspondante est transmise de manière transparente à l'objet serveur qui supporte la méthode invoquée. Le client a seulement besoin de connaître l'interface offerte par l'objet qu'il veut utiliser. Les mécanismes découlant de la distribution et de l'hétérogénéité d'implantation (la localisation de l'objet serveur, le langage de programmation utilisé pour son implantation et le système d'exploitation de la machine hôte de l'objet serveur) sont masqués au client et directement pris en charge par l'ORB.

#### **2.4.2 Comparaison MMS/CORBA et les avantages de l'approche CORBA**

Il est possible d'établir différentes analogies entre le modèle de communication MMS, et l'architecture CORBA. On peut ainsi comparer les services de communications offerts par la pile MAP à ceux d'un ORB. Tout comme le fait le noyau de communication de l'ORB, la pile MAP garantit l'interopérabilité entre des équipements hétérogènes. Néanmoins, MAP repose uniquement sur des services traitant les communications entre un client et un serveur MMS.

De surcroît, elle n'offre pas, contrairement à CORBA, des fonctions étendues permettant de gérer la désignation des ressources distantes et le contrôle des actions concurrentes.

Les services MMS implantés dans un serveur constituent pour un client l'interface d'accès à un VMD. Ces services sont définis dans le langage de spécification ASN.1 (*Abstract Syntax Notation 1*) associé à la couche présentation de la spécification MAP. Ce langage, utilisé par l'ensemble des protocoles d'application définis par l'ISO, permet de décrire les structures de données échangées entre le client et le serveur MMS. De ce point de vue, l'ASN.1 peut être considéré comme un IDL associé à l'environnement ISO. Ainsi, dans notre approche, les services MMS décrits en ASN.1 doivent être transformés en des spécifications en IDL CORBA. On préserve ainsi les spécifications de la norme ISO-MMS. L'avantage est de bénéficier d'un nouvel environnement d'exécution réparti ajoutant des services de transparence d'accès et de localisation qui n'existe pas dans MMS.

Le profil MAP définit une pile de communication adaptée aux applications industrielles. L'avantage de notre démarche, fondée sur CORBA, est de conserver les apports fonctionnels de la norme MMS : interaction client/serveur, machine abstraite de fabrication, hétérogénéité des équipements de production, tout en bénéficiant d'un véritable environnement de développement et d'exécution adapté aux applications distribuées.

Les protocoles TCP/IP sont utilisés pour l'échange de messages à partir de la spécification CORBA 2.0. Ces protocoles sont maintenant implantés efficacement, bien maîtrisés, et largement répandus dans le monde industriel. Cet aspect assure une certaine pérennité à l'approche que nous avons choisie. La prise en compte de l'hétérogénéité des matériels, des systèmes d'exploitation, et la garantie de l'interopérabilité entre différents ORB par le protocole IIOP, correspondent aux besoins d'ouverture nécessaires au déploiement d'un protocole d'application comme MMS.

Le modèle d'interaction client/serveur de MMS correspond au mécanisme d'appel de procédure à distance et peut se traduire par des invocations de méthodes d'objets serveurs CORBA. La sémantique d'invocation des méthodes à distance choisie pour CORBA dans le cas synchrone est fondée sur une sémantique au plus une fois. Elle offre ainsi un certain contrôle en cas de pannes, alors que la tolérance aux pannes n'est pas du tout abordée par la norme MMS. Le modèle d'implantation de l'appel de procédure distant de CORBA propose en outre un mécanisme de remontée d'exception, permettant ainsi un traitement transparent et efficace des pannes ou des erreurs.

CORBA permet aussi une véritable structuration des applications sous la forme d'objets répartis coopérants. Les objets CORBA offrent un moyen d'implanter naturellement la notion de VMD propre au serveur MMS. De plus, par rapport à une implantation OSI de la messagerie industrielle, on bénéficie des mécanismes de désignation des objets qui, à l'aide d'un service de noms, masquent la répartition et la localisation des objets. L'implantation de MMS au-dessus d'un environnement CORBA s'intègre à une démarche globale de conception objet, depuis la phase de spécification jusqu'à la phase d'implantation et d'intégration.

L'approche objet présente un réel intérêt dans le monde des applications industrielles : abstraction, modularité, évolutivité, réutilisation.

Un autre avantage significatif apporté par CORBA pour le portage de MMS réside dans son langage de description d'interface IDL. Dans le cadre de notre approche, l'IDL permet de décrire rapidement et simplement l'interface accessible, en termes d'objets et de méthodes, des différents modules de notre architecture OO-MMS sur CORBA : le client, le serveur, et éventuellement l'application client et l'équipement. CORBA fournit les moyens d'une ingénierie orientée composants aux applications d'informatique industrielle.

Enfin, il n'est plus nécessaire de développer certains services MMS comme la gestion d'événements, la gestion de sémaphores, et la gestion du téléchargement /sauvegarde de domaines, on peut utiliser les services communs CORBA correspondant aux événements, à la concurrence d'accès, et à la persistance. On peut indiquer toutefois l'expérience de transferts vidéo avec ISO-MMS au-dessus d'une pile MAP [SEM99], qui utilise le service de téléchargement et montre l'intérêt d'un tel service indépendamment de la gestion de la persistance. Ce type d'expérience pourrait être très intéressant à mener dans le contexte d'une solution de type OO-MMS sur CORBA.

Tous ces avantages ont été pris en compte directement dans la conception de notre deuxième génération de messagerie industrielle orientée objet.

## **2.5 Conception d'une Messagerie Industrielle Orientée Objet sur CORBA : OO-MMS**

L'architecture OO-MMS que nous présentons dans ce paragraphe, est une extension à l'environnement CORBA des résultats obtenus lors de l'utilisation du mécanisme d'appel de procédure à distance [LEF95a]. Comme pour ce prototypage, nous avons retenu les objets MMS les plus couramment utilisés dans les applications industrielles : le VMD, la variable, le domaine et l'invocation de programme.

### **2.5.1 Adaptation du modèle de communication**

Il est nécessaire d'adapter le modèle de communication associé à la norme MMS à celui de CORBA. Cet aménagement de la norme MMS, entraîne notamment la disparition de la notion d'association (connexion au niveau application entre un client et un serveur MMS), des objets MMS de portée association, de la négociation des paramètres d'environnement, de la gestion des requêtes concurrentes, de l'apparition d'une sémantique d'invocation de méthodes et de la désignation explicite des VMD.

#### **2.5.1.1 Disparition de l'Association**

Le modèle de communication de la norme MMS est fondé sur le mode connecté : un client MMS qui veut accéder aux objets d'un VMD distant doit établir une association à l'aide du service *Initiate*, puis la libérer avec le service confirmé *Conclude* ou le service non confirmé *Abort*. Dans le cadre de l'architecture CORBA, la connexion entre un client et un serveur n'est pas gérée explicitement par la partie applicative, elle est partie intégrante des mécanismes internes de l'ORB. Le mécanisme qui s'approche le plus de la mise en relation d'un client avec un serveur à travers une association correspond, dans l'environnement CORBA, à l'édition de liens qui s'effectue lorsque le client recherche l'objet qui implante la méthode avec lequel il va

interagir. Cette opération retourne une référence d'objet. Ce lien n'est accessible qu'à travers cette référence. La notion d'association telle qu'on la trouve dans la norme MMS disparaît donc avec l'approche que nous avons adoptée.

Or, l'établissement de l'association entre un client MMS et un serveur MMS s'accompagne de la négociation d'un certain nombre de paramètres, principalement la liste des services MMS supportés par le serveur. On suppose que les caractéristiques d'un réseau opérationnel en informatique industrielle sont déterminées à l'avance, et qu'aucune négociation n'est nécessaire. Pour retrouver un caractère dynamique à la mise en correspondance d'un client avec un serveur, on pourrait supposer l'utilisation d'un courtier, entité qu'on désigne par *Trader* dans CORBA [SEI96] et qui correspond aux pages jaunes.

La disparition de l'association a d'autres conséquences. Les objets de portée association n'existent plus et n'ont pas d'équivalent dans une implantation de MMS au-dessus de CORBA. Dans une architecture OSI, un serveur peut être composé d'un ou plusieurs VMD, chacun étant désigné par une adresse de présentation (PSAP, *Presentation Service Access Point*). L'absence d'association engendre le besoin de désigner explicitement le VMD auquel s'adressent les services MMS. Un objet serveur CORBA peut être référencé par une suite de noms hiérarchiques du type arborescence de fichiers. Ce schéma peut s'appliquer à la désignation des modules d'un serveur MMS, et en particulier pour les VMD qu'il contient.

Remarquons toutefois que nos dernières générations de messagerie industrielle, OpenTAZ puis AspectTAZ présentées dans la deuxième partie du chapitre 3, voient ré-apparaître l'association pour matérialiser et gérer les ressources affectées à une relation entre un client et un serveur.

### 2.5.1.2 Asynchronisme des échanges

Le modèle de communication MMS est fondé sur un schéma asynchrone : sur une association donnée entre un client et un serveur, le client peut envoyer plusieurs requêtes, sans attendre la réponse de la première requête. Ainsi, plusieurs requêtes peuvent être en cours de traitement au niveau d'un client. Afin de faire le lien entre une requête et une réponse, chaque requête est caractérisée par un identificateur de lancement *InvokeID*.

Un service MMS confirmé est implanté dans CORBA à l'aide d'une invocation de méthode synchrone. Un identificateur de lancement n'est donc pas nécessaire, le lien entre une requête et une réponse étant pris en charge par l'ORB. Par contre, le synchronisme des appels CORBA interdit *a priori* le traitement en parallèle de plusieurs requêtes chez un client. Afin de conserver les caractéristiques de MMS, il est donc nécessaire d'utiliser plusieurs activités parallèles (processus légers ou *multi-threading*) pour le traitement des requêtes concurrentes au niveau d'un client.

Indépendamment du client, le traitement parallèle des requêtes côté serveur dépend des capacités intrinsèques du système d'exploitation supportant l'ORB. Quand le système d'exploitation implante le *multi-threading*, le parallélisme d'exécution du client se prolonge

naturellement vers le serveur. Nous obtenons cette propriété sur le prototype OO-MMS décrit dans [GUY97a] [GUY97b] [GUY02].

### 2.5.1.3 Traitement des données non sollicitées

La plupart des services MMS sont confirmés. Toutefois, la norme MMS permet au serveur d'envoyer spontanément des données non confirmées vers le client, les deux principaux services étant *UnsololicitedStatus* pour transmettre l'état logique et l'état physique du VMD, et *InformationReport* pour transmettre la valeur d'une variable. Cette capacité à traiter les données suivant un modèle événementiel est essentielle pour les applications de productique : gestion de seuils et d'alarmes. Il nous semble donc important de les prendre en compte dans une implantation de MMS au-dessus de CORBA. Nous avons choisi de les mettre en oeuvre en utilisant les invocations synchrones sans retour de résultat. Les premiers prototypes utilisaient des invocations unidirectionnelles dites *oneway* [GUY97a] [GUY97b] [GUY02] moins intéressants par rapport à la recherche de propriétés de fiabilité. En effet, la sémantique des invocations unidirectionnelles est "peut-être" tandis que la sémantique des invocations synchrones est "au plus une fois" avec génération d'exception en cas d'anomalie. Cette situation est préférable pour transporter des données de type alarmes, puisque c'est l'usage que nous en faisons dans notre prototype.

### 2.5.2 Implantation des services MMS

Dans le cadre du modèle ISO-MMS, un client désigne explicitement un VMD pour accéder aux abstractions MMS qu'il contient. L'interface de ce VMD est constituée par l'ensemble des services MMS implantés au niveau du serveur.

Nous adoptons un schéma analogue pour l'implantation des services MMS sur une architecture CORBA : un VMD est un objet CORBA dont l'interface d'accès correspond à l'ensemble des services effectivement implantés (figure 5.). Cet objet est obtenu à partir de la description en IDL des services MMS suivant le principe décrit dans la partie 2.5.3.

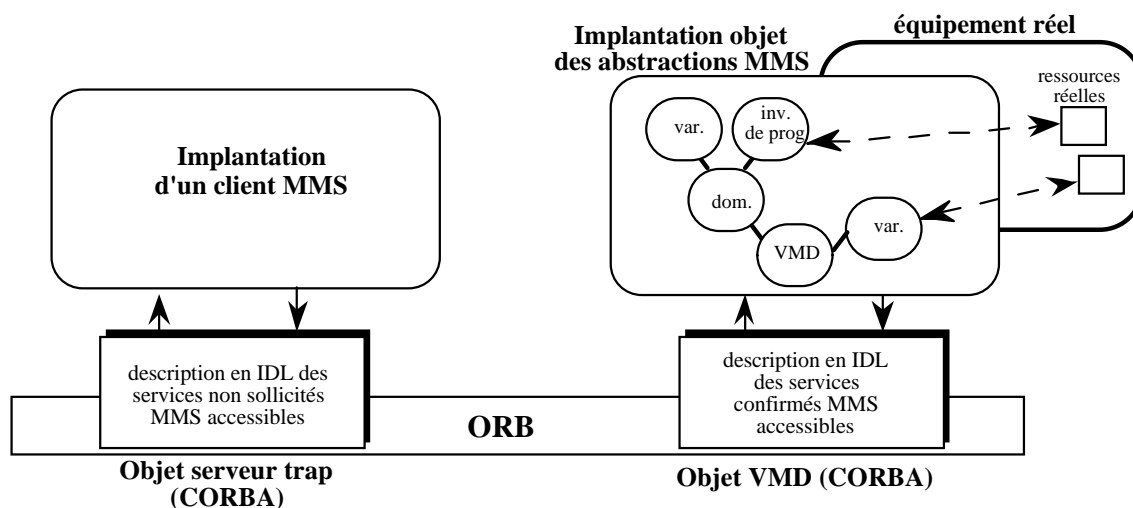


Figure 5. Principe général d'implantation des services MMS

Avec notre approche, l'ORB est utilisé pour accéder à l'objet VMD distant, en bénéficiant des mécanismes de communication et de désignation propres à CORBA. Lorsqu'un client

demande l'exécution d'un service MMS, le service de désignation recherche la référence de l'objet VMD correspondant et l'adaptateur d'objet invoque la méthode et retourne les résultats au client.

Les abstractions MMS que nous considérons (VMD, variables, domaines, invocations de programme) sont implantées au dessus de l'objet CORBA VMD, en tant qu'objets applicatifs écrits dans le langage de programmation de l'implantation. Ces abstractions respectent l'organisation hiérarchique des objets d'un VMD définie dans la norme MMS : une variable peut être contenue dans un VMD ou un domaine, le code exécutable d'une invocation de programme est localisé dans un domaine... Leurs attributs sont conformes à ceux définis dans la norme [ISO91a]. Le rôle des abstractions MMS est de prendre en compte la requête émanant de l'objet VMD, de faire le lien avec une ressource de l'équipement réel et de retourner éventuellement les paramètres de réponse attendus. L'implantation des abstractions MMS doit donc respecter la signature des opérations définies dans la description IDL de l'objet VMD.

L'objet VMD permet d'implanter les services confirmés. Or, la norme ISO-MMS autorise également un serveur à envoyer des données à un client, par le biais des services non sollicités. Dans le cadre de l'architecture CORBA, nous implantons ces services à l'aide d'un objet CORBA dit serveur trap, défini côté client. Cet objet contient la signature en IDL des services non confirmés envoyés par le serveur, déclarés en tant que méthodes synchrones sans retour de résultats.

Avec cette approche, nous respectons les principes fondamentaux de la norme ISO-MMS. Le client a seulement besoin de connaître l'interface de l'objet VMD, c'est à dire les paramètres du service qu'il utilise. Le lien entre l'abstraction MMS et la ressource de l'équipement est pris en charge par l'implantation du serveur. Il s'agit d'un problème local qui n'est pas traité par la norme ISO-MMS. Pour gérer le lien entre OO-MMS et les parties strictement applicatives (client et équipement) nous proposons d'utiliser le bus logiciel. Cette solution présente plusieurs avantages : elle masque les spécificités des mécanismes de communications locales (mémoire partagée, files de messages, tubes...), le bus logiciel CORBA utilisé peut implanter les communications locales très efficacement, elle oblige une spécification claire des interfaces des parties applicatives, elle permet une approche composant de bout en bout de la chaîne contrôle-commande à distance.

### **2.5.3 Passage d'une spécification ASN.1 à une spécification IDL**

Comme nous l'avons indiqué dans l'introduction, nous cherchons à porter MMS sur une nouvelle architecture. Cet objectif nécessite la transposition de la norme MMS. Comme tous les protocoles de la couche application du modèle OSI, une grande partie de la norme MMS, notamment pour les unités de données de protocole ou PDU (*Protocol Data Unit*), utilise le langage de description ASN.1. En dehors de l'OSI, l'ASN.1 n'a pas été véritablement adopté. Ainsi, quand on cherche à porter MMS hors de l'environnement OSI, un travail de traduction dans le langage de spécification de l'environnement cible est indispensable.

Il faut prendre en compte la structuration apportée par les services dans le travail de transposition des PDU MMS spécifiées en ASN.1. Dans notre solution, une méthode dans l'interface d'un serveur CORBA correspond à un service MMS. La PDU de requête définit le paramètre d'appel de la méthode, la PDU de réponse précise le paramètre de retour. Les PDU d'erreur sont traduites en exceptions.

Pour illustrer notre démarche, nous présentons un exemple de traduction de ASN.1 en IDL avec le service confirmé *Identify* qui est associé à l'objet VMD. Notons qu'avec la disparition de l'association (cf. paragraphe 2.5.1.1), certains éléments décrits en ASN.1 ne sont pas utilisés, ils ne sont donc pas traduits et n'apparaissent pas dans la spécification IDL. Cet aspect est général à toutes les traductions opérées pour les différents services.

Le service *Identify* va donc définir une opération de l'interface de l'objet CORBA VMD gérant les services confirmés. Comme toute norme du niveau application, MMS introduit une différence entre la primitive de service et les PDU générées par le service. Cette différence se résume principalement à rajouter un identificateur de lancement *InvokeID*. Dans la mesure où cette information n'est plus nécessaire dans une approche CORBA, on peut directement mettre en correspondance les paramètres des primitives de service et le coeur des PDU qu'elles génèrent. Pour la requête (*Identify.req*) envoyée par le client, *Identify-Request* étant vide, la méthode *Identify* dans la spécification IDL ne possède pas de paramètre d'appel. Pour la réponse (*Identify.resp*), le serveur MMS envoie une PDU qui contient sous forme de chaînes de caractères le nom du fabricant, le nom et la version du modèle de l'équipement, ce qui est décrit en ASN.1 par :

```
Identify-Response ::= SEQUENCE {
    vendorName      [0] IMPLICIT VisibleString,
    modelName       [1] IMPLICIT VisibleString,
    revision        [2] IMPLICIT VisibleString
}
```

qui se traduit en IDL immédiatement par :

```
struct IdentifyResponse {
    string vendorName;
    string modelName;
    string revision;
};
```

En raison de l'état du VMD ou de l'équipement, qui peut être par exemple non opérationnel, le serveur MMS peut ne pas pouvoir rendre le service demandé. Il envoie dans ce cas, une réponse négative avec une PDU d'erreur, *ConfirmedErrorPDU*, en expliquant la raison. Les PDU d'erreur des services MMS ont toutes le même coeur qui correspond au type *ServiceError*. Ce type générique a donc une structure très complexe. Cette description intervient dans la spécification IDL de l'opération *Identify* pour définir une exception que doit traiter le client MMS. Comme nous préférons avoir un traitement global des exceptions pour

tous les services MMS que nous implantons, nous avons conservé la combinatoire de la spécification ASN.1 de *ServiceError* :

```

ServiceError ::= SEQUENCE {
    errorClass [0] CHOICE {
        vmd-state [0] IMPLICIT INTEGER
        {
            other (0),
            vmd-state-conflict (1),
            vmd-operational-problem (2),
            domain-transfer-problem (3),
            state-machine-id-invalid (4)
        },
        application-reference[1] IMPLICIT INTEGER
        {
            other (0),
            application-unreachable (1),
            connection-lost (2),
            application-reference-invalid (3),
            context-unsupported (4)
        },
        ...
    }
}

```

Sa traduction en IDL se fait en plusieurs étapes. On transforme d'abord chacun des types d'erreur en énumération IDL. Puis on élabore une structure de données équivalente au *ServiceError* en ASN.1 pour discriminer les différents types d'erreurs et pour pouvoir définir en IDL l'exception *ServiceError*. On obtient alors les définitions suivantes :

```

enum CategorieVmdState{
    otherVmdState,
    vmdStateConflict,
    vmdOperationalProblem,
    domainTransferProblem,
    stateMachineIdInvalid
};

enum CategorieApplicationReference{
    otherApplicationReference,
    applicationUnreachable,
    connectionLost,
    applicationReferenceInvalid,
    contextUnsupported
};

```

```

...

enum TypeError {
vmdState,
applicationReference,
...
};

union ServiceErrorType switch(TypeError) {
case vmdState          : CategorieVmdState  Valeur_vmdState;
case applicationReference : CategorieApplicationReference
                          Valeur_applicationReference;
...
};
exception ServiceError {
ServiceErrorType serviceErrorVal;
};

```

On en déduit la spécification IDL de l'opération Identify dans l'objet CORBA VMD chargé des services confirmés :

```

interface serveurMMS {
...
IdentifyResponse Identify() raises (ServiceError);
...
};

```

Le résultat obtenu par une transcription à la main est assez proche d'une transcription automatique dont les éléments sont donnés au paragraphe 2.6.

## 2.6 Prototypages d'une Messagerie Industrielle Orientée Objet

### 2.6.1 Architecture d'un prototype OO-MMS

Le prototype OO-MMS s'est construit en trois étapes : la spécification des services MMS en IDL et la mise en oeuvre de ces services, puis le test des fonctionnalités implantées dans OO-MMS. Cette dernière correspond à la réalisation de l'environnement utilisateur côté client, et à l'équipement côté serveur. Cette dernière étape est spécifique à l'architecture que nous proposons, elle n'est pas décrite dans la norme MMS. Elle correspond à la mise en place des mécanismes d'interactions avec les parties applicatives et à une approche d'ingénierie logicielle orientée composant. La portabilité du code des objets MMS qui permet leur migration est obtenue dans le prototype [LAU99] en utilisant Java comme langage de programmation qui permet un certain affranchissement des systèmes d'exploitation et du matériel.

La figure suivante donne une vue d'ensemble de l'architecture. Pour une description plus précise, nous renvoyons le lecteur à [GUY02].

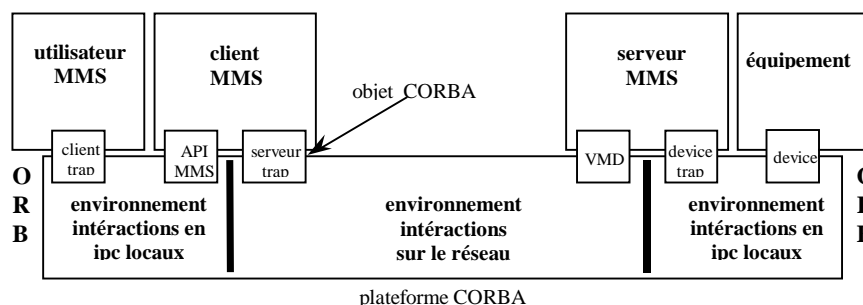


Figure 6. Architecture d'un prototype OO-MMS

## 2.6.2 Les Différents Prototypes

Plusieurs prototypes ont été réalisés, on peut distinguer deux familles. La première utilise le bus logiciel à objet CORBA CHORUS/COOL de Sun Microsystems (Chorus Systems a été racheté par Sun Microsystems vers la fin 1997). Ils sont programmés en C++. Un premier prototype utilise Unix [SIM96] [GUY97a]. Le second est implanté sur le micro-noyau ChorusOS [GUY97b] [GUY02], et bénéficie de ses caractéristiques temps réel et multi-threads. A partir de la description IDL de l'interface de l'objet VMD, le compilateur CHIC de COOL (IDL vers C++) fournit le module souche et le module squelette d'invocation statique. Ces modules contiennent les fonctions d'interfaçage avec l'ORB et la traduction en C++ des méthodes de l'objet VMD. Le premier prototype sur Unix n'implantait que la gestion du VMD, et la gestion des variables. Le second prototype ajoute la prise en compte des domaines et des invocations de programmes. Cette famille de prototypes implantait les services non-sollicités avec des méthodes unidirectionnelles.

La seconde famille de prototypes a été programmée en Java [LAU99] et utilise différents ORB : ORBacus [LAU98], Jonathan [DUM98]. Elle fonctionne sur différentes plateformes : Windows95, et Linux en particulier. Elle sert de base au projet COCA [GRE98]. Compte tenu de la date à laquelle ces prototypes ont été mis en œuvre, ils utilisent tous le BOA (Basic Object Adapter). Toutefois, avec une plate-forme utilisant une version plus récente de CORBA (à partir de 2.2), et donc le POA (Portable Object Adapter), nous n'aurions pas besoin des sophistications apportées par cette nouvelle fonctionnalité, ce qui nous laisserait envisager des configurations telles que Minimum CORBA [OMG02b].

La spécification IDL du premier prototype a été traduite de l'ASN.1 vers l'IDL CORBA manuellement, conformément aux principes décrit dans le paragraphe 2.5.3 en éliminant les champs inutiles à cause de l'environnement CORBA. La spécification pour les prototypes en Java résulte d'une traduction automatique ASN.1 vers IDL [SMI97] générée par un chercheur de l'INRIA, Monsieur Nader Soukouti, et transmise par l'EPFL [MER98]. Cette traduction automatique a été aussi expurgée des champs inutiles. A notre grande surprise, le résultat des deux traductions est finalement assez proche pour les services que nous avons implantés.

### 2.6.3 Projet COCA

Il a pour objectif de prouver l'adéquation de notre approche à un cas réel. Ce projet a été conduit en commun avec l'équipe GRPI de l'IUT de St Denis, avec R. Boissier en tant que chef de projet de la partie commande-numérique.

Le projet COCA implante un contrôle à distance d'une machine à commande numérique dans un atelier de fabrication de pièces de type PME. Il vise un second objectif qui est de proposer une approche orientée composant dans la conception des applications d'informatique industrielle de types contrôle-commande ou supervision. Cet objectif découle de l'utilisation d'un bus logiciel orienté objet. Le projet s'inspire de la norme d'accompagnement pour les commandes numériques MMS-NC [ISO91b] et intègre les parties messagerie industrielle et les parties applicatives, en particulier l'architecture commande numérique ouverte, OPEN'NC [BOI97], définie par l'équipe GRPI de l'IUT de St Denis. Le projet COCA a aussi fait émerger de premières contraintes temps réel [BOI98].

La machine à commande numérique est une fraiseuse 3 axes Wirth & Gruffah gérée par un contrôleur PMAC Delta Tau. Les programmes du pilote de périphérique sont écrits en C et ont été portés de DOS à Linux, ils sont accessibles par les programmes Java au travers du mécanisme de méthodes natives Java [EPI98]. L'ensemble des composants reliant l'utilisateur au pilote de commande numérique fonctionne. Il est programmé en Java, et utilise ORBacus [LAU98] [OOC99] sur Linux.

## 2.7 Conclusion

Nous avons présenté la conception et le prototypage d'une messagerie industrielle orientée objet à partir d'une approche bus logiciel de type CORBA. Cette messagerie, OO-MMS prend comme modèle la norme MMS [ISO90] [ISO91a] de l'ISO. Ce travail est original, et résulte de l'observation des besoins utilisateurs en informatique industrielle. Ce projet prend sa source dans le besoin de transplanter MMS hors de la pile OSI. Les principaux avantages sont: intégration de l'approche objet, démarche système réparti, ingénierie des applications de productique orientée composants.

La messagerie industrielle orientée objet obtenue est assez démonstrative. L'architecture proposée est valide, et a même retenu l'attention du groupe de travail Manufacturing Domain Task Force, puisque ce travail a été présenté sur invitation à l'OMG lors de la réunion de Burlingame en Septembre 2000 [GRE00a]. Ce travail a par ailleurs fait l'objet d'extensions [ARI98] [ARI01].

Marginalement, on peut noter que dans le cadre de OO-MMS, nous n'avons pas exploité complètement les apports de CORBA. Notre travail n'a pas mis en oeuvre les services de gestion d'événements (si on adopte un point de vue différent de l'équipe espagnole en essayant de ré-utiliser les services de CORBA de façon littérale), de contrôle de concurrence, et de persistance. C'est un approche qui évite l'implantation des abstractions événements et sémaphore de la norme ISO-MMS, et peut faciliter les actions de téléchargement et de sauvegarde de domaines. Il faut remarquer toutefois que le service de téléchargement peut être intéressant à conserver, [SEM99] utilise le télé-chargement pour intégrer à l'aide de ISO-MMS un transfert de données multi-média aux flux de contrôle-commande.

Toutefois, on peut dresser un bilan sur un plan différent. En effet, d'un point de vue ingénierie logicielle, plusieurs prototypes ont été développés. Si on englobe les travaux sur les RPC, on voit qu'on a effectué plusieurs fois le même travail de prototypage, sur deux plates-formes de type différent (CORBA et ONC-RPC) en utilisant différents langages de programmation (C, C++, Java). Même si chaque nouveau prototype a pu bénéficier de l'expérience acquise avec ses prédécesseurs, on peut chercher quelle approche permettrait de développer la messagerie industrielle une fois et qu'elle puisse s'exécuter sur différentes plates-formes. Par exemple, nous n'avons pas essayé ni RMI, ni DCOM, et nous ne sommes pas en mesure d'utiliser .NET ou SOAP/XML sans tout re-développer.

Le choix de l'ORB Jonathan [OBJ02] [DUM98] a été une première tentative de réponse à ce besoin, en effet, Jonathan dispose d'une personnalité CORBA et d'une personnalité RMI. Compte tenu du gel des évolutions de Jonathan à cette époque, nous avons cherché une autre approche. Récemment, nous avons fondé nos travaux sur la programmation orientée Aspects [KIC97] [KIC01]. Les développements actuels de la messagerie industrielle orientée objet s'effectue en Java et s'appuie sur l'environnement JAC [AOP02]. Ce n'est plus tout à fait la même messagerie industrielle puisque nous avons remplacé ISO-MMS par TASE.2, mais le problème à résoudre est identique. Ce travail a été effectué en coopération avec Lionel Seinturier du LIP6.

L'approche Jonathan est maintenant développée spécifiquement sous la forme d'un canevas d'aide à l'assemblage de composants logiciels programmés en Java, commercialisé par la société KELUA. Les deux approches nous semblent prometteuses. Toutefois, nous explorons l'utilisation de la technologie JAC [TEB02a] [TEB02b] qui est plus directement accessible au CEDRIC à l'heure actuelle. Il reste toutefois un travail important de comparaison de ces deux approches pour évaluer laquelle des deux est la plus appropriée pour l'ingénierie des services systèmes distribués en informatique industrielle, ou si elles sont complémentaires.

## 3 Vers une Messagerie Industrielle Orientée Objet Temps Réel

### 3.1 Introduction

Le chapitre précédent a montré la conception et le prototypage d'une messagerie industrielle orientée objet sous plusieurs formes, et a démontré la viabilité de notre approche. Toutefois, nous sommes à la moitié du chemin décrit dans l'introduction. Pour obtenir une messagerie industrielle temps réel, il faut travailler sur deux axes : l'environnement support et la messagerie elle-même. Dans les deux cas, il faut les enrichir avec des capacités temps réel. La suite des travaux décrit porte sur l'étude de ces capacités temps réel.

Dès le début du projet SAFFRES et tout naturellement, la prise en compte de capacités temps réel des plates-formes s'est manifestée à travers les expérimentations sur ChorusOS, un micro-noyau temps réel du commerce, doublé d'une vraie plate-forme système distribué. Une proposition a été faite par exemple pour que le modèle MMS sur RPC présenté Figure 4 (c) fonctionne au-dessus de l'ipc de ChorusOS [GRE96]. Le choix de COOL comme premier ORB dans nos expériences pour le modèle figure 4 (d) plutôt qu'un ORB disponible sur Internet répond à ce besoin. Enfin, dans le cadre de SAFFRES, un important travail a été fait autour de la cohérence des données afin d'offrir une mémoire répartie partagée temps réel, nous avons exploré les mécanismes offerts par ChorusOS: mapper [COR95] [THI97] [BON96] [BOL97] [BOU97] [SAU99], et surtout proxy-mapper [BEC97]. Le proxy-mapper permet d'installer dans le micro-noyau les services de gestion d'une cohérence, l'appel à ces services est alors très efficace puisque le code service est exécuté par le thread appelant sans commutation de contexte grâce à un appel de procédure léger (trans-espace d'adressage utilisateur-noyau). Ce mécanisme de mémoire répartie partagée a été abandonné car il ne convenait pas à au support d'objets cohérents, le grain de gestion de la cohérence étant la page un conteneur trop gros et inapproprié [MOL99]. En fait le proxy-mapper couplé à un noyau parfaitement profilé pour du temps réel et un réseau à QoS dans une configuration telle que celle qui a été définie dans les travaux de C. Lizzi dans sa thèse [LIZ99c] pourrait permettre la construction d'une mémoire répartie partagée équivalente à une mémoire réfléchie [VME96] en logiciel, donc adressant des contraintes temps réel moins strictes. L'étude de propriétés temps réel avec ChorusOS a vraiment été poussée au maximum avec la thèse de C. Lizzi qui a intégré ATM et ses propriétés de Qualité de Service au micro-noyau en le modifiant pour gérer des contraintes d'échéances (deadline) en réparti. Les travaux de C. Lizzi ont montré l'efficacité d'une approche sur mesure intégrant la QoS et la mise en œuvre de propriétés temps réel à tous les niveaux d'une plate-forme répartie du réseau jusqu'à l'ORB, ici Java.

La prise en compte de contraintes temps réel au niveau de la messagerie industrielle ISO-MMS a fait l'objet d'un certain nombre d'études, [ROD90] [ELL95] [AKA95] par exemple. Quelques soient les propositions, elles n'ont pas emporté l'adhésion de la communauté. Nous avons choisi de travailler à partir d'une norme d'accompagnement d'ISO-MMS qui s'appelle TASE.2 (Telecontrol Application Service Element version 2) connue aux Etats-Unis sous le

nom ICCP (Inter-control Center Communication Protocol) et normalisée par l'IEC (retrouver le sigle). TASE.2 spécifie un service de gestion de variables avec des propriétés temporelles (date de production de la variable, fourniture périodique ou sur condition ...). Ce choix présente plusieurs avantages :

- IEC-TASE.2 spécifie un comportement temporel ajouté aux éléments de messagerie industrielle définis dans ISO-MMS,
- c'est surtout une norme, et donc elle peut faire l'adhésion d'une communauté d'utilisateurs et de fournisseurs de produits,
- enfin, cette spécification, nous rapproche d'un service d'échange de machine à machine pour la construction d'une base de données répartie temps réel et de donc de notre objectif posé en 1997.

Ce chapitre se découpe en deux parties. La première partie, 3.2, décrit notre expérience de l'extension de la messagerie industrielle orientée objet sur ATM, elle donne un cadre conceptuel fondé sur le modèle RM-ODP étendu par les travaux du consortium ReTINA [BLA97]. La partie suivante, 3.3 décrit nos travaux relatifs à IEC-TASE.2 sur CORBA en C++, dénommé OpenTAZ[OPE02]. Cette deuxième série de travaux n'est pas achevée. La partie 3.3.3 décrit les résultats, extensions et perspectives plus en détails. La partie 3.4, décrit la poursuite dans plusieurs directions : la programmation orientée aspects dont le prototype AspectTAZ [TEB02a] [TEB02b] représente un premier essai, et, un projet d'utilisation de messagerie industrielle dans une application multimédia, le Concert Réparti.

### **3.2 Conception d'une Messagerie Industrielle Orientée Objet Portable à QoS sur ATM**

La conception d'une Messagerie Industrielle Orientée Objet Portable (on peut dire aussi migrable à cause de l'utilisation Java, ce qui a un sens pour la partie client mais pas tellement pour le serveur si les équipements industriels n'ont pas une partie opérative avec des interfaces et des services identiques) à QoS adressait initialement le domaine de la fabrication de pièces avec des commandes numériques.

Dans le domaine des machines à commande numérique[ISO91b] [PAP96], les contraintes temps réel portent essentiellement sur les délais de transfert d'un client vers un équipement, sur des temps de réponse, les contraintes de débit sont relativement faibles [BOI98]. Pour adresser les contraintes temps réel, une première approche consiste à d'étendre le standard MMS de l'ISO avec des propriétés temps réel [ROD90] [AKA95] [ELL95] [MAA97]. De nouvelles approches tentent d'utiliser une approche CORBA temps réel [KUN98] [POL98]. La suite du projet consiste à mettre en oeuvre une plateforme capable de prendre en compte des contraintes temporelles telles que celles décrites précédemment. Ces contraintes nécessitent de nouvelles infrastructures, aussi bien en terme de réseaux que de systèmes, pour pouvoir être satisfaites.

Le premier objectif dans notre projet concerne la prise en compte de contraintes temps réel au niveau des interactions entre objets de la messagerie industrielle tout en préservant les

résultats acquis lors de la phase précédente, c'est à dire en conservant les bénéfices d'une approche objets répartis, et d'une certaine portabilité offerte par l'utilisation de Java.

Nous avons choisi ATM car c'est le seul réseau qui implante des propriétés de QoS de façon native et facilement accessible au niveau API sur Linux. ATM est accédé à travers la boîte à outils AJAX [SEI99], et, un bus logiciel modulaire et extensible Jonathan écrit en Java lui aussi [DUM98]. Jonathan et AJAX permettent de spécifier les paramètres de QoS sans avoir à modifier la signature des services associés aux objets de la messagerie obtenus avec OO-MMS, cet avantage n'est atteint par aucune autre solution dont nous avons connaissance.

Il faut noter que l'association de MMS et d'ATM conjointement a fait l'objet d'une certaine attention par la communauté scientifique, on peut citer deux projets qui s'y sont intéressés dans des contextes différents. Une implantation de MMS sur ATM existe et est décrite dans [MAA97]. [WU96] utilise MMS pour le contrôle-commande de périphériques multimédia qui communiquent en utilisant un réseau ATM.

Pour prendre en compte complètement des contraintes temps réel applicatives, il faut un environnement d'exécution adapté sur chaque machine, nous envisageons l'utilisation d'un micro-noyau temps réel durci [LIZ99a] utilisant ATM [LIZ98a] [LIZ98b] et de machines virtuelles Java temps réel [NIL96]. L'utilisation d'ATM apporte un avantage supplémentaire car il permet d'intégrer des flux de données multimédia au flux de données de contrôle-commande [BAC99?].

Dans ce paragraphe, nous fournissons une première solution en présentant l'intégration du prototype OO-MMS sur une architecture réseaux à qualité de service de type ATM. Cette intégration nécessite quelques adaptations aux concepts et à l'environnement d'exécution CORBA envisagé aux paragraphes précédents. Pour cela, le paragraphe 3.2.1 présente un modèle d'ORB flexible qui facilite de telles extensions. Le paragraphe 3.2.2 décrit alors l'architecture AJAX qui intègre des liaisons de données ATM à des communications entre objets CORBA. Finalement, le paragraphe 3.2.3 présente le nouveau prototype QoS-OO-MMS supporté par une plateforme CORBA-like au-dessus d'un réseau de communication ATM et dresse un premier bilan sur ses performances.

### **3.2.1 Approche ORB flexible pour la qualité de service**

#### **3.2.1.1 Modèle ReTINA**

Le projet ReTINA [TIN95] a eu pour but de développer un environnement de programmation distribuée et flexible pour les services de télécommunications. L'un de ses principaux objectifs est de permettre le développement d'applications à contraintes temps réel fortes telles que les applications multimédia. Son architecture s'appuie sur le modèle de référence RM-ODP [BLA97] et son implantation sur la norme CORBA.

Deux abstractions majeures du modèle RM-ODP sont présentes dans ReTINA : la notion d'objet et celle d'objet de liaison. En particulier, un objet de liaison (voir exemple dans la figure ci-dessous) est une entité composite, distribuée entre plusieurs espaces d'adressage appelés capsules, et mettant en œuvre les mécanismes nécessaires aux interactions entre

objets. Un objet de liaison représente donc une liaison complète de bout en bout. Il fournit une transparence à la localisation lors des accès et gère les ressources réseau (protocoles, modes de communication, formats de données) et système (souches logicielles, politiques de gestion des tampons et des activités).

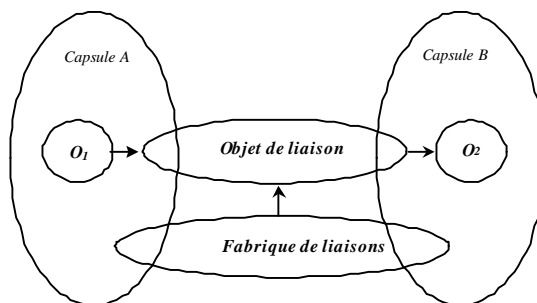


Figure 7. Exemple de liaison point à point (d'après [DUMANT 98])

Les objets de liaisons sont mis en place par des fabriques de liaison. Chaque fabrique est spécialisée pour un type de communication. Par exemple, on peut envisager des fabriques pour des communications TCP/IP au-dessus d'une architecture Ethernet, des fabriques pour des liaisons Java RMI, ou encore, des fabriques pour des liaisons CORBA au-dessus d'une architecture ATM. Lorsque des garanties temporelles sont mises en jeu par les liaisons comme c'est le cas dans ce dernier exemple, leur gestion est entièrement du ressort de l'objet de liaison. Ces derniers encapsulent donc les fonctionnalités du réseau, aussi bien au niveau des couches basses que des couches hautes. Ils sont vus comme des fournisseurs de services par les objets applicatifs. Une des caractéristiques de l'architecture ReTINA réside dans la possibilité de définir et de faire cohabiter au sein d'une même application différents types de liaisons ayant chacune leur sémantique propre. De plus, le concept de fabrique fournit des environnements flexibles, au sens où les objets implantant ces fabriques peuvent être chargés, installés ou retirés dynamiquement en cours d'exécution.

Une notion importante du modèle ReTINA est celle d'interface. Les interfaces permettent aux objets d'interagir et sont désignées par des références. Ces dernières sont des structures de données concrètes qui fournissent le type de l'interface référencée et un ensemble d'identificateurs permettant de la désigner de manière non ambiguë (ce sont par exemple des références CORBA ou des UUID DCE). Les interfaces sont particulièrement pratiques pour la spécification de contraintes de QoS.

Lorsque des contraintes temps réel sont nécessaires aux applications, les objets de liaison les gèrent par l'intermédiaire d'équations de qualité de service. Selon le modèle défini dans [BLA97], la liaison est alors vue comme un flux de données et est associée à une suite d'invocations de méthodes. Chaque invocation (voir figure 8) se décompose en événements d'émission de l'invocation (EI), de réception de l'invocation (RI), d'émission de la réponse (ER) et de réception de la réponse (RR). Les équations de QoS sont exprimées en logique temporelle QL (Qos Language) [BLA97] et traduisent des contraintes temporelles entre les événements précédents. Les équations sont par exemple, de la forme suivante :

$$d_1 \leq t(e, n+k) - t(e', n) \leq d_2$$

où  $e$  et  $e'$  sont deux événements,  $t(evt, n)$  la date de la  $n$ -ème occurrence de l'événement  $evt$  et  $d_1$  et  $d_2$  des unités de temps. Une telle relation stipule que la  $n$ -ème occurrence de l'événement  $e'$  doit être de la  $(n+k)$ -ème occurrence de l'événement  $e$ , d'au moins  $d_1$  et d'au plus  $d_2$  unités de temps.



Figure 8. Événements associés à une liaison à QoS

Cette approche fournit un modèle général qui permet de spécifier simplement les contraintes associées à chaque liaison. La figure qui suit donne un exemple dans le cas d'échanges de données périodique pour la messagerie industrielle TASE.2 décrite au paragraphe 3.3.

```
// TASE.2 server object
interface <operational> DataSetTransferSetManagement {
    ...
    void startDataSetTransferSet (...);
    ...
}

// TASE.2 client object
interface <stream> TransferReportManagement {
    ...
    void transferReport (...);
    ...
}

Provided clause
time(TransferReportManagement.transferReport.SendEvent,1) –
    time(DataSetTransferSetManagement.startDataSetTransferSet.SendEvent,1) = 500 ms
and
∀n time(TransferReportManagement.transferReport.SendEvent,n+1) –
    time(TransferReportManagement.transferReport.SendEvent,n) = 400 ms
Required clause
time(DataSetTransferSetManagement.startDataSetTransferSet.ReceiveEvent,1) –
    time(DataSetTransferSetManagement.startDataSetTransferSet.SendEvent,1) ≤ 250 ms
```

Figure 9. Spécification de contraintes temps réel pour des envois périodiques toutes les 400ms, l'envoi périodique commençant 500ms après l'invocation du client.

### 3.2.1.2 Le bus logiciel Jonathan

Jonathan [DUM98], développé par le CNET, est une implantation en Java du modèle ReTINA. C'est essentiellement un cadre logiciel ("*framework*") qui, par le biais des notions d'objet, d'interface d'objet et de fabrique de liaisons, permet de mettre en place des environnements de programmation distribuée. Une fabrique pour des liaisons CORBA au-dessus de TCP/IP est fournie en standard faisant ainsi de Jonathan un bus logiciel conforme à la norme CORBA 2.0.

La figure 10 décrit l'architecture de la fabrique de liaisons CORBA implantée par Jonathan. GIOP est le protocole générique défini par les spécifications de l'OMG pour acheminer les invocations de méthodes entre les objets client et serveur. Il requiert un protocole de transport fiable. Dans la majorité des cas, TCP est utilisé. La version de GIOP pour TCP est alors désignée par l'acronyme IIOP ("*Internet Interoperable Object Protocol*"). Jonathan gère un mécanisme de connexions persistantes qui permet de multiplexer plusieurs invocations de méthodes IIOP successives sur une même connexion TCP. Du côté client, les souches préparent les messages d'appel et traitent les messages de retour, tandis que du côté serveur, les squelettes assurent le traitement et l'aiguillage des requêtes vers les serveurs et préparent les messages de retour.

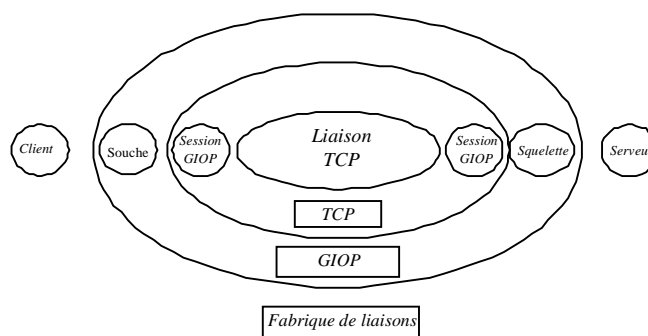


Figure 10. Structure des liaisons CORBA IIOP (d'après [DUM98])

La modularité de Jonathan facilite l'intégration de types de liaisons variés pour des communications entre objets distribués. Par exemple, une fabrique pour des liaisons multipoints via le protocole multicast-IP est fournie par le CNET et le support des interactions conforme au protocole RMI de Sun est en cours de développement. Le respect des équations de qualité de service présentées au paragraphe précédent dépend des interfaces et des caractéristiques du réseau sous-jacent. Par exemple, dans le cadre de la fabrique de liaisons ATM présentée au paragraphe suivant, ces équations doivent être traduites en terme de débit et de gigue pour pouvoir être transmises lors des procédures d'ouverture de connexion.

### 3.2.2 L'environnement AJAX

Ce paragraphe décrit la fabrique de liaisons ATM qui a été développée pour Jonathan. Elle permet de mettre en œuvre la messagerie OO-MMS présentée au paragraphe 2.2.5 en tirant parti de la qualité de service offerte par le réseau ATM sous-jacent. Nous commençons par

rappeler quelques éléments sur la qualité de service dans les réseaux ATM (paragraphe 3.2.2.1), puis nous présentons l'architecture de la fabrique de liaisons (paragraphe 3.2.2.2).

### 3.2.2.1 Qualité de service dans les réseaux ATM

La technologie ATM introduit, en plus de débits élevés, des garanties sur les caractéristiques des communications. Différents ensembles de garanties, appelés classes de service, sont définis dans les spécifications ATM [ATM96a] [ATM96b]. Le but de ce paragraphe n'est pas de les décrire exhaustivement, mais de résumer leurs caractéristiques principales (pour de plus amples informations, le lecteur peut se référer par exemple, à [ALL95]). Chaque classe de service est destinée à un type de trafic particulier et est associée à un certain nombre d'attributs de qualité de service. A un instant donné, en fonction de la charge du réseau, l'admission ou le rejet d'une liaison possédant ses caractéristiques propres est réalisé à l'ouverture de la connexion par une phase de négociation. Actuellement, six classes de service sont définies dans les réseaux ATM :

- CBR ("Continuous Bit Rate") : fournit une liaison de données à débit constant. Parmi tous les attributs de qualité de service utilisable, deux ont ici une importance particulière. Ce sont PCR ("Peak Cell Rate") et CDV ("Cell Delay Variation") qui définissent respectivement, le débit et la gigue tolérable pour cette liaison.
- VBR ("Variable Bit Rate") : fournit une liaison dans laquelle le débit peut varier. Deux sous-classes pour respectivement, des trafics temps réel (rt-VBR), et non temps réel (nrt-VBR), sont proposées. Le paramètre de qualité de service BT ("Burst Tolerance") associé à cette classe définit l'amplitude des pics de trafic admissibles.
- ABR ("Available Bit Rate") : en fonction de la charge du réseau et des mécanismes de contrôle de flot d'ATM, la bande passante allouée à l'utilisateur de cette classe de service peut varier entre une valeur minimale, définie par l'attribut MCR ("Minimum Cell Rate") et une valeur maximale, définie par l'attribut PCR ("Peak Cell Rate").
- GFR ("Garanteed Frame Rate") : reproduit un service de type Frame Relay sur ATM avec un débit garanti CIR (Committed Information Rate), et un débit en surplus EIR (Excess.... Information Rate)
- UBR ("Unspecified Bit Rate") : les liaisons de données de ce type utilisent au maximum la bande passante disponible, mais aucune garantie n'est fournie sur le trafic. Cette classe se comporte donc de façon identique à une liaison Ethernet.

L'unité d'information de base véhiculée par les réseaux ATM est la cellule (composée de 53 octets dont 5 d'en-tête). Quatre protocoles (les AALs "ATM Adaptation Layer") sont disponibles pour le transport des unités de données applicatives. Numérotés AAL1, AAL2, AAL3/4 et AAL5, ils sont destinés à différents types de trafic. Néanmoins, AAL5 est le protocole le plus performant et est presque systématiquement utilisé, quel que soit le trafic.

### 3.2.2.2 Architecture de la fabrique de liaisons AJAX

AJAX (acronyme de "Atm for Jonathan And linuX") est, au sein de l'ORB Jonathan, une fabrique permettant à des objets CORBA de communiquer par l'intermédiaire de liaisons ATM. L'encodage des données, les souches clientes et les squelettes de serveurs créés par

cette fabrique, sont identiques à ceux des liaisons CORBA habituelles. Par contre, la pile protocolaire IIOP/TCP est remplacée par le couple AIOP/AAL5.

Le protocole AIOP ("Atm Interoperable Object Protocol") que nous avons conçu, fonctionne au-dessus de AAL5 et permet de véhiculer des invocations de méthodes entre deux objets CORBA. AIOP gère les connexions ATM de manière persistante : la même connexion peut être réutilisée pour des invocations successives entre deux objets CORBA identiques. AIOP fournit les fonctionnalités suivantes :

- gestion des invocations de méthodes et de leurs retours (dans le cas d'invocations synchrones),
- mise en œuvre d'un mécanisme de fragmentation/réassemblage pour les messages dont la taille est supérieure à la limite de 65535 octets imposée par AAL5,
- levée d'exception en cas de perte de données ou d'erreurs de communication.

Par ailleurs, le protocole AIOP modifie le mécanisme de référence d'objet, appelé IOR ("Interoperable Object Reference") de CORBA. Un IOR permet de localiser de manière unique tout objet dans un environnement réparti de type CORBA. Nous avons adapté ce mécanisme afin de substituer aux adresses IP utilisées par les protocoles IIOP/TCP, des adresses ATM NSAP.

### **3.2.2.3 Prototype QoS-OO-MMS : OO-MMS à qualité de service**

Nous décrivons, dans ce paragraphe, l'implantation de la version objet de l'environnement MMS à qualité de service. Cette version étend les résultats décrits au paragraphe 5.3. Elle a été réalisée pour le système d'exploitation Linux, avec une architecture de réseaux ATM et l'ORB Jonathan.

#### **3.2.2.3.1 Caractéristiques du prototype**

Notre prototype utilise la librairie "ATM on Linux" [ALM97] qui fournit une interface de programmation identique à celle des sockets BSD pour les connexions ATM. Les données sont transportées par le protocole AAL5 fonctionnant au-dessus de circuits virtuels ATM commutés. Deux classes de services sont disponibles : UBR et CBR (voir paragraphe 5.6.2.1). Ce dernier est associé à une interface de qualité de service permettant de spécifier un débit, une gigue et la taille des paquets AAL5. Les deux premiers paramètres correspondent aux attributs PCR et CVD de la classe de service CBR. Une classe Java native fournit l'interface avec les primitives de cette librairie écrite en C. Au-dessus de cette classe, le protocole de transport AIOP décrit au paragraphe 3.2.2.2. Des souches clientes et serveurs pour gérer des connexions ATM sont fournies. Elles peuvent être héritées par les classes de notre implantation objet de MMS. Il est néanmoins important de noter que cette architecture n'est pas dépendante de MMS mais peut être utilisée pour tout type d'applications fonctionnant sur CORBA.

Les objets MMS communiquent par appels de méthodes. Ceux-ci sont véhiculés par une connexion CBR qui garantit un débit constant. Une nouvelle connexion est ouverte, de manière transparente, dès qu'un nouveau couple d'objets MMS souhaite communiquer. Nous

fournissons pour chaque objet, une interface avec une méthode *setQoS* et deux paramètres : un débit en bits/s sous la forme d'un entier long, et une gigue en microsecondes sous la forme également, d'un entier long. Ces paramètres sont utilisés à l'ouverture de la connexion. Ils ne peuvent, actuellement, être renégociés dynamiquement. Néanmoins, ils pourront l'être lorsque nous mettrons à niveau notre plate-forme en passant de la version 3.1 à la version 4.0 [ATM96b] du protocole de signalisation UNI utilisé par ATM.

### 3.2.2.3.2 Plate-forme d'expérimentation et tests de performance

Notre plate-forme d'expérimentation comprend deux PC Pentium Pro à 200 Mhz munis de cartes ATM Efficient Network 155Mbits/s reliés par un commutateur. Le système d'exploitation utilisé est Linux 2.1.x, et l'environnement de développement, le JDK 1.1.x de Sun. Des tests de performance montrent que, jusqu'à des débits de l'ordre de 10 Mbits/s entre deux objets MMS Java, la liaison CBR fournit un service satisfaisant : une dizaine de liaisons de ce type peuvent être ouvertes simultanément.

Néanmoins, un certain nombre de problèmes apparaissent pour des débits demandés supérieurs à 10 Mbits/s. Ils sont essentiellement dus à la sémantique de la classe de service CBR. Bien que cette dernière fournisse un débit constant, le protocole de transport AAL5 ne garantit pas la livraison des paquets. Ainsi, des données peuvent être perdues si par exemple, le récepteur ne lit pas les données assez rapidement. Des tests de performances [ALM97] effectués par les auteurs de la librairie "ATM on Linux" montrent que pour un débit demandé de 130 Mbits/s entre deux PCs connectés directement, un taux de perte de 13% peut être observés pour un simple programme client/serveur écrit en C. Nos propres tests montrent que ce taux est atteint dès 80 Mbits/s lorsque les PCs sont connectés via un commutateur, et dès 15 Mbits/s lorsque les programmes écrits en Java sont interprétés par la machine virtuelle du JDK. Deux causes principales justifient ce résultat :

- le surcoût d'exécution introduit par la machine virtuelle Java,
- l'existence de deux zones de mémoire cloisonnées (une pour la machine virtuelle et une pour le code natif C) qui empêche la mise en place d'une politique mémoire de type zéro-copie.

A titre de comparaison, l'architecture sur-mesure pour faire de la QoS de Service proposée par C. Lizzi [LIZ99c] donne les performances suivantes : la communication entre deux threads Java sur deux machines différentes (sans technique Just In Time) offre un débit de 97Mb/s et un délai de 15ms (à travers un seul commutateur) avec un taux de perte de 0.3% (les pertes sont liées à la gestion de buffer par au réseau) sur un lien ATM à 155Mb/s. Ces performances s'expliquent par les propriétés de l'architecture : le noyau effectue l'ordonnancement sur la base d'échéances et de critères de criticité, il prend en compte l'héritage d'échéance de bout en bout sur les communications inter-processus et sur les verrous d'exclusion mutuelle, l'ORB est construit à partir d'une JVM domaine public étendue avec les capacités temps réel du noyau étendu et du réseau ATM, enfin, la gestion mémoire est optimisée sur un schéma 0-copie de l'espace carte réseau jusqu'à la JVM avec l'interface JNI (Java Native Interface) en passant par le micro-noyau.

### 3.2.3 Conclusion

Ce paragraphe montre la conception d'une messagerie industrielle orientée objet utilisant un réseau à QoS. L'architecture résultante nous semble correcte, toutefois, elle n'atteint pas un niveau de performance satisfaisant pour des applications d'informatique industrielle:

Pour le type d'objectif que nous visions, on peut suspecter que l'utilisation de Java et d'une JVM standard est un premier obstacle. On peut se poser la question sur l'effet d'une JVM temps réel pour se rapprocher des conditions du prototype de C. Lizzi. Nous n'avons pas pu mener ce type d'investigation. Toutefois, l'expérimentation d'une JVM temps réel du commerce, PERC de Newmonics [NIL96] [NEW00], s'est montré prometteuse [REV01a]. Pour la suite du projet, le prototypage s'est effectué en C++ afin de limiter le nombre de composants intervenant dans la plate-forme.

ATM était un bon choix pour se familiariser avec la QoS, mais ce type de réseau n'a pas réussi à s'affirmer pour les communications de machine à machine, et n'est pas prêt de pénétrer le domaine de l'informatique industrielle. Une expérience sur l'accès à un réseau ATM via différents systèmes d'exploitation ne s'est pas montrée concluante [RIZ00]. Il reste une technologie de cœur de réseau, donc invisible généralement à l'application qui utilise plutôt les protocoles TCP/IP. Actuellement, il n'y a pas de technologie capable de rivaliser avec ATM. Deux approches se présentent dans l'univers Internet : IntServ et DiffServ. L'approche IntServ se réduit aujourd'hui à évaluer l'adéquation du couple technologie MPLS (Multi-Protocol Label Switching) et protocole de réservation de ressource RSVP-TE [PUJ01] (ReSource reserVation Protocol Traffic Engineering) à notre besoin. L'approche DiffServ vise une garantie de QoS statistique, mais offre de nombreuses combinaisons dans le choix des classes et le paramétrage des réseaux. C'est l'approche la plus en vogue, mais elle nécessite une infrastructure de gestion de QoS [PUJ00] [PUJ01] qui n'est pas encore déployée sur les campus universitaires. Pour les deux approches, l'usage de TCP pour des applications temps réel n'est pas ce qu'il y a de mieux : le contrôle d'erreur, et les mécanismes anti-congestion (slow-start et congestion avoidance) introduisent délai et gigue dans les transferts. Il serait préférable d'avoir des solutions fondées sur UDP et RTP/RTCP.

La thèse de C. Lizzi a montré qu'on pouvait obtenir d'excellentes performances vis à vis des applications d'informatique industrielle pourvu qu'on puisse fabriquer une plate-forme répartie temps réel "sur-mesure". Le succès de cette plate-forme fait aussi sa faiblesse [LIZ00], le résultat même s'il est probant est trop spécifique. Aujourd'hui, pour être crédible vis-à-vis de partenaires industriels, il faut développer ses résultats sur des plates-formes assemblées à partir de composants sur étagère.

Finalement, les perspectives d'un travail visant à fournir une messagerie industrielle orientée objet sur une plate-forme temps réel répartie doivent nécessairement prendre en compte tous les niveaux de la plate-forme : réseau des couches réseau à l'ORB en passant par le système d'exploitation et ses propriétés temps réel [BAC00] [GRE00b] [GRE01b]. Nos prochaines investigations devraient porter sur CORBA-RT [OMG02c] au-dessus d'un noyau temps réel, et sur l'influence des approches IntServ/DiffServ pour la gestion de priorités distribuées.

### 3.3 Une messagerie industrielle orientée objet avec des caractéristiques temps réel

Cette deuxième partie adressant des objectifs temps réel s'appuie sur la norme de messagerie industrielle TASE.2 [IEC96a] [IEC96b] [KEM96] [EPR98] de l'IEC. IEC-TASE.2 est moins complexe que ISO-MMS car le nombre d'abstractions est moindre. IEC-TASE.2 peut être vue comme la spécification d'un service d'échange de données pour une mémoire répartie partagée d'objets de types variables (des mesures plus exactement) construit sur un modèle producteur-consommateur avec des paramètres de QoS temporelle associés aux variables. L'intérêt principal vis-à-vis des objectifs de la HDR d'IEC-TASE.2 est que cette norme ajoute des caractéristiques temporelles aux abstractions d'une messagerie industrielle. Cet aspect nous a beaucoup intéressé car le travail résultant est issu d'une norme donc plus apte à être accepté par la communauté informatique industrielle.

#### 3.3.1 Caractéristiques Principales de TASE.2

Le protocole IEC-TASE.2 est une norme d'accompagnement de ISO-MMS. IEC-TASE.2 est en fait bâti au-dessus de ISO-MMS, et se sert donc des abstractions de celle-ci. La vocation première de IEC-TASE.2 est de supporter des échanges de données de production entre une centrale électrique et un centre de régulation de la production. Plus globalement elle peut servir aussi dans le contrôle de la production et de la distribution d'eau ou de gaz. Aujourd'hui, avec la libéralisation du marché de l'énergie, IEC-TASE.2 pourrait servir de base à une application de négoce d'énergie entre producteurs et consommateurs au niveau européen comme PowerNext [url à retrouver].

IEC-TASE.2 est décrit comme un protocole Client/Serveur. Deux types d'interactions sont définis : les opérations initiées par les clients correspondent à des requêtes/réponse retournant un résultat, et les actions initiées par les serveurs correspondent à des notifications avec des données. Quatre types de transferts sont fournis : "exactement une fois" (interaction client/serveur classique), "périodique" (émission périodique par le serveur), "type exception" (transfert sur changement d'état), "type conditionné" (transfert sur changement de condition). Les deux derniers types peuvent être assimilés. Une caractéristique importante du modèle IEC-TASE.2 est que les échanges de données entre un client et un serveur IEC-TASE.2 sont conditionnés par l'existence d'une "association", au sens ISO du terme mais enrichie avec des aspects fonctionnels, qui doit être explicitement créée lors d'une phase d'initialisation. Cette association est un objet de communication utilisé pour englober des propriétés de contrôle d'accès et des paramètres de QoS sous la forme d'attributs relatifs aux variables TASE.2. L'association est couplée à une Table Bilatérale qui définit formellement l'ensemble des données pouvant être échangées à travers l'association ISO entre un client et un serveur.

Les fonctions de IEC-TASE.2 sont découpées en 9 blocs de conformité. Lors de l'évaluation de IEC-TASE.2 pour EDF [BEC01a] [BAC99e], nous avons travaillé sur les blocs 1, 2 et 5. Le Bloc 1 définit un ensemble minimal de services lié à l'échange de données périodiques. Le Bloc 2 étend le bloc 1 en offrant un service d'émission de données conditionnelles souvent appelé "Report By Exception". Pour l'implantation sur un ORB, seuls les blocs 1 et 2 ont été concernés dans nos études.

IEC-TASE.2 définit des objets "Data Value" et "Data Set" (supportés par des variables nommées ou des listes de variables nommées de ISO-MMS) gérés par un serveur. Il y a plusieurs catégories de Data Set, nous nous sommes intéressés aux Data Set Transfer Set qui regroupent des Data Value. Généralement, les Data Value d'un même Transfer Set sont fonctionnellement corrélés du point de vue de l'application. La gestion des Data Value et des Data Set [KEM96] est relative à la consultation d'objets existants, leur création, leur destruction ... Les Data Values réfèrent dans la pratique des points de mesure, "Indication Points". Les points de mesure peuvent contenir des informations relatant un statut sous la forme d'une suite de bits, une valeur analogique entière ou réelle. On associe à ces mesures des attributs comme une estampille temporelle, une indication de qualité par rapport à son domaine de définition (par exemple "valide", "invalid", "affectée par opérateur", "suspecte"...), le nombre de changements de valeur depuis la dernière récupération... La figure 11 donne la définition d'un point de mesure dans IEC-TASE.2. Le tableau ci-après donne les différents types de points de mesure dans TASE.2.

```

Object: IndicationPoint
  Key Attribute: PointName
  Attribute: PointType (REAL, STATE, DISCRETE)
  Constraint PointType=REAL
    Attribute: PointRealValue
  Constraint PointType=STATE
    Attribute:PointStateValue
  Constraint PointType=DISCRETE
    Attribute: PointDiscreteValue
  Attribute: QualityClass: (QUALITY, NOQUALITY)
  Constraint: QualityClass = QUALITY
    Attribute: Validity (VALID, HELD, SUSPECT, NOTVALID)
    Attribute:CurrentSource(TELEMETERED,CALCULATED,ENTERED,ESTIMATED)
    Attribute:NormalSource(TELEMETERED,CALCULATED,ENTERED,ESTIMATED)
  Attribute: NormalValue (NORMAL,ABNORMAL)
  Attribute: TimeStampClass: (TIMESTAMP, NOTIMESTAMP)
  Constraint: TimeStampClass = TIMESTAMP
    Attribute: TimeStamp
    Attribute: TimeStampQuality: (VALID, INVALID)
  Attribute: COVClass: (COV, NOCOV)
  Constraint: COVClass = COV
    Attribute: COVCounter

```

Figure 11. Spécification des objets TASE.2 Indication Point

Type Mesures	Simple	+ code qualité	+ estampille temporelle	+ compteur COV
Statut	Data_State	Data_StateQ	Data_StateQTimeTag	Data_StateExtended
Entier	Data_Discrete	Data_DiscreteQ	Data_DiscreteQTimeTag	Data_DiscreteExtended
Réel	Data_Real	Data_RealQ	Data_RealQTimeTag	Data_RealExtended

Table 1. Différents Types de Points de Mesure considérés dans TASE.2

Attributs Spécifiés		Signification	
<b>EventCodeRequested</b>		Code spécifique d'événement applicative.	
<b>DS Transmission Pars</b> (conditions de génération de compte rendu par le serveur)	<b>Interval (1)</b>	Période séparant l'émission de deux compte-rendus.	
	<b>Start Time (1)</b>	Date d'activation de la période.	
	<b>RBE, Report By Exception (2)</b>	faux: comportement périodique, tous les objets d'un Data Set Transfer Set sont envoyés dans le compte-rendu, vrai: comportement événementiel, seuls les objets du Transfer Set qui ont changé sont envoyés dans le compte-rendu.	
	<b>Integrity Check (2)</b>	Intervalle définissant une échéance pour envoyer l'ensemble des valeurs des objets d'un Data Set Transfer Set qu'ils aient changé ou non, lié à la condition Integrity Time Out.	
	<b>TLE, Time Limit for Execution (2)</b>	Echéance pour générer un compte-rendu et l'émettre.	
	<b>Buffer Time (2)</b>	Durée pendant laquelle on garde des changements de valeur d'un objet avant de générer un compte-rendu, la dernière valeur produite seule est envoyée, lié à la condition Object Change. Dans le domaine de la production d'énergie, on appelle cette valeur "bande morte".	
	<b>Critical (2)</b>	Les client doivent accuser réception des compte-rendu.	
	<b>DS Condition Requested</b> (indicateurs autorisant la génération de compte- rendus)	<b>Interval Time Out (1)</b>	Active l'envoi de compte-rendus contenant toutes les données d'un Data Set Transfer Set en fin de période, corrélé à Interval et RBE faux.
		<b>Interval Time Out (2)</b>	Active l'envoi de compte-rendus contenant uniquement les données d'un Data Set Transfer Set qui ont changé en fin de période, corrélé à Interval et RBE vrai
		<b>Operator Request</b>	Active l'envoi de compte-rendus sur demande opérateur.
<b>Object Change (2)</b>		Active l'envoi de compte-rendus sur changement de valeur, corrélé à Buffer Time et RBE vrai	
<b>Integrity Time Out (2)</b>		Active l'envoi d'un compte-rendu contenant tous les objets d'un Data Set Transfer Set quand l'intervalle de temps, spécifié dans le paramètre Integrity Check, expire.	
	<b>Other External Event</b>	Active l'envoi de compte-rendus sur événements applicatifs	

(1) paramètre lié au bloc fonctionnel 1, RBE a la valeur faux, (2) paramètre lié au bloc fonctionnel 2, RBE a la valeur vrai

Table 2. Attributs des Data Set Transfer Set spécifiant les conditions de génération d'un Transfer Report

Les Data Set Transfer Sets spécifient des conditions de production des compte-rendus, Transfer Reports, vers un client. La signification des paramètres les plus représentatifs est donnée dans le tableau ci-dessous. La figure 12 donne un aperçu de la gestion des variables dans TASE.2.

## Client TASE.2

## Serveur TASE.2

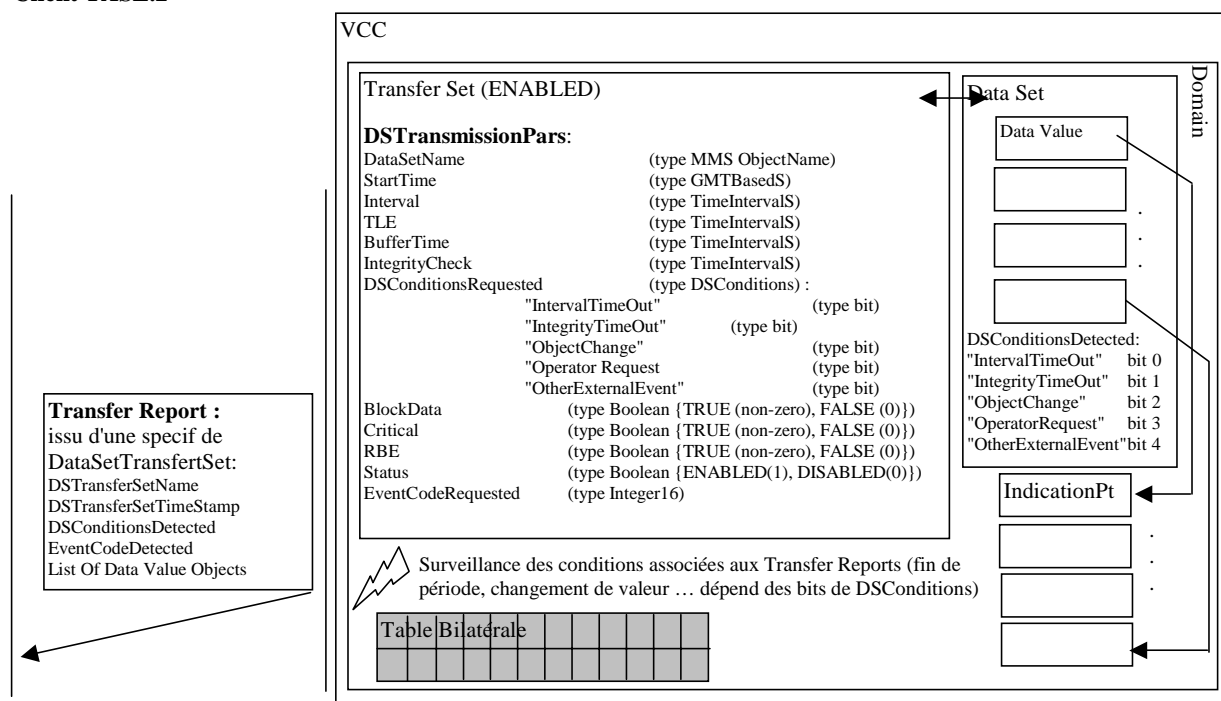


Figure 12. Vue Schématique de la Gestion des Variables dans TASE.2

Notre première expérience dans l'équipe avec la norme IEC TASE.2 a été un portage d'une souche produit vers un micro-noyau temps réel [BEC01b] effectué par E. Becquet dans le cadre de sa thèse. L'étude demandée par EDF R&D s'est transformée en une véritable évaluation du produit avec l'équivalent d'une campagne de tests de conformité. Cette expertise a aussi servi à l'évaluation fonctionnelle de la norme TASE.2 pour les besoins du producteur d'énergie.

### 3.3.2 OpenTAZ, l'implantation objectifiée de IEC-TASE.2 sur CORBA

OpenTAZ est un projet de type logiciel libre qui implante les fonctions de la norme IEC TASE.2 au-dessus de CORBA. Le prototype correspondant s'appuie sur MICO [MIC02]. Il est écrit en C++.

Nous avons deux choix de conception possibles :

- respecter le principe d'une norme d'accompagnement, c'est à dire implanter notre transposition de IEC-TASE.2 au-dessus de OO-MMS, choix suggéré dans [GRE01a],
- adopter la transposition de IEC-TASE.2 sur CORBA directement.

La deuxième option a été retenue pour des raisons de simplicité et de souci de performance. Par ailleurs, elle permettait de mieux adhérer au modèle RM-ODP/ReTINA comme nous le verrons par la suite.

Comme pour les travaux relatés dans le paragraphe 3.2, la conception de OpenTAZ se fonde sur le modèle RM-ODP/ReTINA qui est tout à fait adapté à la représentation des interactions entre objets, et qui offre un canevas rigoureux pour la spécification de contraintes

temporelles sur ces interactions. Dans la transposition de IEC-TASE.2 à CORBA, on retrouve un grand nombre des principes de la transposition de ISO-MMS à CORBA. Le VMD est remplacé par le (VCC, Virtual Control Center), on retrouve un serveur CORBA de chaque côté (rôle client et rôle serveur IEC-TASE.2). Les abstractions rattachées à un VCC sont implantées comme des objets locaux et sont par conséquent programmées comme dans OO-MMS.

Les différences viennent de l'IDL, et de la représentation des liaisons entre les rôles client et serveurs IEC-TASE.2. Cette partie clef a été très fortement imprégnée de la culture RM-ODP/ReTINA grâce à la participation de F. Horn (anciennement membre de l'équipe Architecture des Systèmes et des Réseaux de France Telecom R&D de J-B. Stefani, et maintenant co-fondateur de KELUA):

- La première partie du travail a consisté à regrouper les différents services en fonction de leur orientation. L'examen de la norme IEC-TASE.2 fait ressortir l'importance de l'abstraction association qui remonte de la couche ISO-ACSE (Association Control Service Element) jusqu'au service spécifié par la norme d'accompagnement à travers la couche ISO-MMS. Toutes les échanges font intervenir explicitement l'abstraction association qui permet de relier les ressources accordées à un client pour ses interaction avec un serveur (la table bilatérale, le contrôle d'accès, les canaux de communication notamment). Vis-à-vis de l'abstraction association, on retrouve donc naturellement un interface de gestion, "Association Management interface", et un interface de contrôle, "Association control interface". Un interface est dédié aux opérations (invocation d'une entité rôle client IEC-TASE.2), et un autre est dédié aux actions (invocation d'une entité rôle serveur IEC-TASE.2 pour les TransferReports). L'ensemble des interfaces définies est schématisé dans la figure ci-dessous.

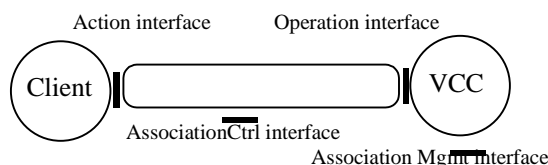


Figure 13. Objet de liaison pour la transposition de IEC-TASE.2 sur CORBA

Le lecteur peut remarquer le côté paradoxal de notre situation. Pour OO-MMS, nous avons décidé de faire disparaître l'association considérant que l'ORB prenait en charge naturellement les tâches dévolues à cette partie. Pour OpenTAZ, elle ré-apparaît. Finalement, c'est la conception d'OpenTAZ qui est la meilleure: l'association dans la pile OSI a un rôle de désignation, d'affectation et de gestion des ressources à une relation entre deux entités applicatives. Cette abstraction et son rôle sont importants dès qu'on cherche à intégrer la QoS dans les interactions, l'association nous avait manqué dans QoS-OO-MMS.

- La spécification IEC-TASE.2 pour sa partie strictement fonctionnelle est quasiment écrite en langue naturelle. Les structures de données sont décrites en pseudo-langage

de programmation structuré comme Pascal. L'essentiel de la construction de la norme repose sur la façon dont les services des différents blocs sont projetés sur ISO-MMS. La spécification de l'IDL des serveurs CORBA a été faite par F. Horn, nous avons discuté quand il y avait des ambiguïtés pour voir comment faire des choix, l'IDL généré a été ensuite validé lors de l'implantation par E. Becquet. Pour les blocs fonctionnels implantés (bloc 1 : échanges de mesures périodiques, ou, bloc 2: échanges sur condition vérifiée) dans le prototype OpenTAZ, cette approche a été très efficace, l'IDL n'a été que très peu retouché.

La transcription d'une partie de la norme IEC-TASE.2 en IDL CORBA est intéressante. La spécification des interfaces et des types devient plus claire, et est indépendante de toute technologie. Cette spécification est sous licence logiciel libre [OPE02]. La figure ci-dessous est un extrait de la spécification relative aux objets Data Value.

```
typedef string DataValueNameType;
typedef sequence < DataValueNameType> DataValueNameSeqType;
exception UnknownDataValueNameType { };
interface DataValueManagement {
    DataValueType getDataValue(in DataValueNameType name)
        raises (UnknownDataValueNameType);
    void setDataValue(in DataValueNameType name, in DataValueType val)
        raises (UnknownDataValueNameType);
    DataValueNameSeqType getDataValueNames();
    DataValueType getValueType(in DataValueNameType name)
        raises (UnknownDataValueNameType);
};
```

Figure 14. Spécification IDL de l'Interface des objets Data Value

Le prototype réalisé par E. Becquet est implanté en Java et reprend les principes d'implantation QoS-OO-MMS tout en introduisant les 4 interfaces citées plus haut. Les interfaces "Operation Interface", "Association Management Interface", "Association Control Interface" sont supportées par le VCC et donc par un objet serveur CORBA. L'interface "Action Interface" est supporté par le client TASE.2, et implante un objet serveur CORBA pour les méthodes qui y correspondent, essentiellement la gestion des TransferReports des DataSetTransfertSet qui peuvent être normale ou oneway en fonction des besoins du client TASE.2 en fonction de l'activation de la clause Critical ou non.

La figure ci-dessous présente un modèle d'utilisation d'OpenTAZ aussi bien pour les communications de type bus de terrain que pour le contrôle de production distant.

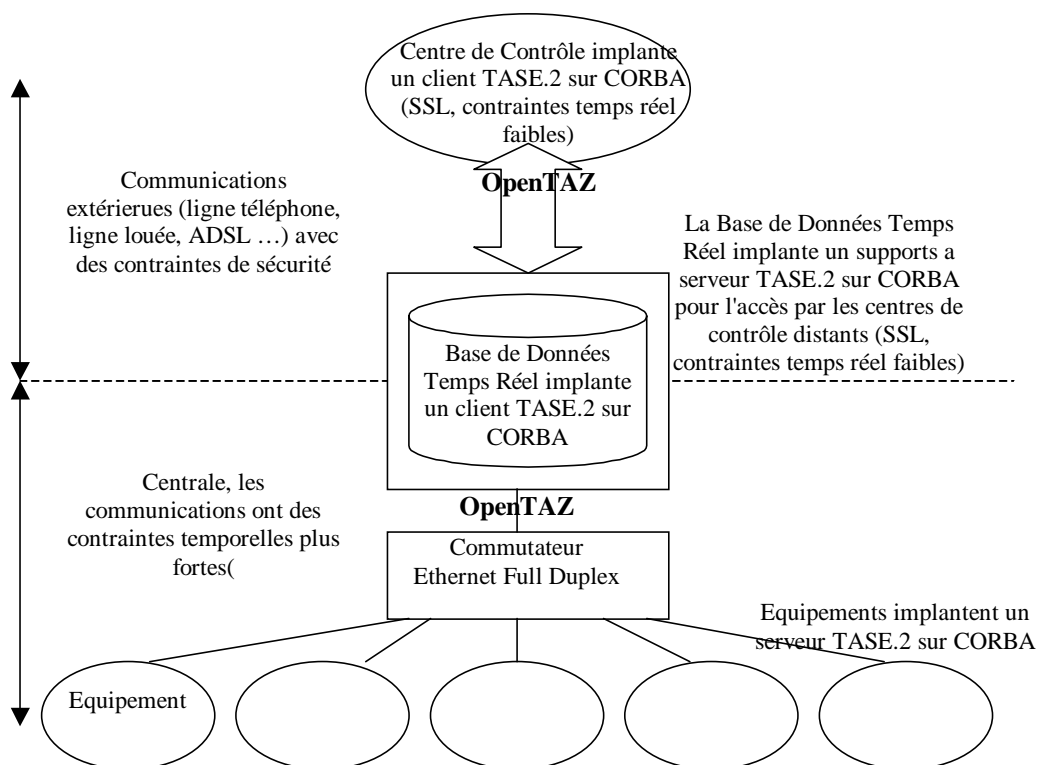


Figure 15. Cas d'utilisation d'OpenTAZ (TASE.2 sur CORBA)

Dans une approche complètement conforme aux travaux CORBA, les équipements pourraient figurer des entités de type Smart Transducers [OMG00].

### 3.3.3 Conclusion

Ce travail est en pleine évolution, et ouvre de nombreuses perspectives dont un certain nombre est exposé dans [rapport de recherche CEDRIC à créer à partir du papier EMSOFT'02] et que nous rappelons ci-après.

#### 3.3.3.1 Emergence d'un protocole de gestion de variables équivalent à un bus de terrain

L'émergence d'un protocole de gestion de variables fonctionnellement équivalent à celui offert par un bus de terrain résulte d'une série d'études avec EDF qui ont permis de définir un contexte d'utilisation de TASE.2 pour les unités de production d'énergie petites et moyennes. L'architecture d'un équipement de surveillance de telles unités est représentée dans la figure ci-dessous. Elle correspond à une solution de même niveau de fonctionnalité que le boîtier proposé par M2MMatrix [retrouver l'URL de Laurent Bacon]. Nos expérimentations ont porté sur un processeur embarqué de type MBX 821, un Power PC de Motorola, avec un micro-noyau temps réel, pSOS+ de Wind River. Le choix de pSOS+ résulte d'une étude antérieure sur l'évaluation de noyaux temps réel pour le compte d'EDF, comme indiqué dans l'introduction de la HDR.

Le processeur est accessible à distance. Il supporte un module de calcul temps réel, un module d'instrumentation et mesure en relation avec un bus de terrain, une base de données temps réel résidente en mémoire qui permet d'accéder aux variables relevées et aux variables calculées sur le procédé, agent d'archivage des alarmes et commandes console, et un serveur

de servlets pour l'IHM déporté via le web comme proposé dans [GUI00]. On suppose que les communications avec l'extérieur sont chiffrées et que tout accès par un utilisateur distant nécessite une authentification préalable. Les utilisateurs distants peuvent être : un opérateur de centrale, des opérateurs de maintenance.

Tel que nous concevons notre solution, les communications extérieures passe par un lien point à point (RTC-modem, GSM, satellite voire ADSL si c'est possible ou pourquoi pas une ligne louée). Cette liaison supporte le protocole TCP/IP. On peut imaginer dans l'absolu la possibilité d'utiliser un réseau VPN. Au-dessus de TCP/IP, deux protocoles d'application sont nécessaires : HTTP, et TASE.2. TASE.2 sert à la gestion de production d'énergie, il est remplacé par OpenTAZ. HTTP permet l'accès via un navigateur, on peut imaginer un accès distant via un PC portable ou via un PC de poche par le personnel de maintenance.

Les communications de centrale entre le processeur embarqué, les senseurs et les actionneurs supposent une solution de base de type Ethernet Industriel [IAO02] et donc avec TCP/IP aussi. Un bus de terrain tel que MODBUS [MOD01] convient parfaitement. Toutefois, les communications en centrale sont aussi spécifiées par une norme, UCA.2 [EPR98b][EPR98c]. De façon synthétique, les protocoles de communication applicatifs sont équivalents à TASE.2 sans le contrôle d'accès. OpenTAZ est prévu aussi pour ce niveau dans notre architecture si le choix n'est pas figé.

Ce projet suppose à terme le développement de processeurs Java pour supporter des capteurs et des actionneurs intelligents. Les cartes Tini [IBU00] représentent un bon exemple de ce type de processeurs. On peut alors imaginer une version de TASE.2 en Java sur ce type de processeur soit utilisant un ORB CORBA ou Java-RMI, voire SOAP si l'utilisateur le demande.

Remarquons enfin, que la réalisation d'une telle application embarquée, bien structurée en modules qui sont complètement identifiés par des fonctions métier, se prêterait parfaitement à une implantation fondée sur une approche de type programmation orientée aspects que nous présentons très brièvement plus loin dans ce chapitre.

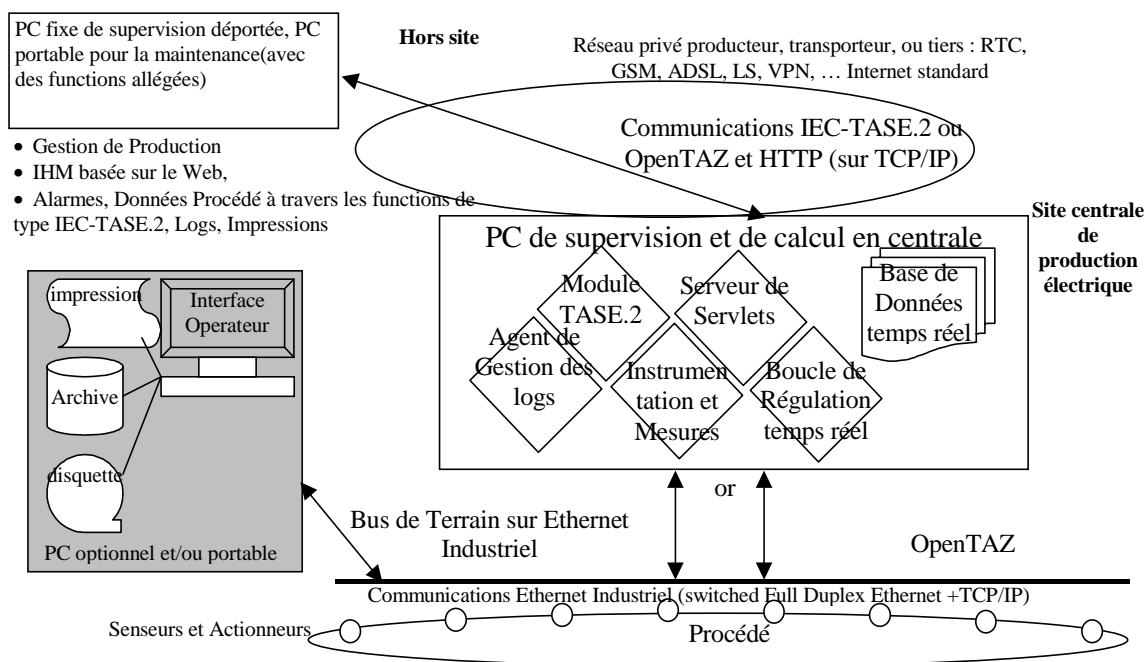


Figure. 16. Architecture d'un boîtier de contrôle-commande et de supervision distants pour les unités de production d'énergie petites et moyennes.

### 3.3.3.2 Conformité partielle au RFP Data Distribution Service

Le Request For Proposal de l'OMG "Data Distribution Service for Real-Time Systems" [OMG01] définit 3 niveaux de conformité. Le travail relatif à OpenTAZ s'intéresse à une conformité minimale, le niveau L0. La conformité est atteinte si le service proposé est conforme au modèle producteur-consommateur, s'il fournit un modèle de donnée et quelques propriétés de base que nous allons voir brièvement au fur et à mesure dans ce paragraphe.

OpenTAZ répond aux spécifications L0 de cette RFP et va un peu au-delà par la structure même de la norme IEC-TASE.2 qui a inspiré nos travaux, spécialement les blocs 1 (échanges périodiques) et 2 (échanges sur occurrence de condition):

- Les DataSetTransferSets et les TransferReports amènent à OpenTAZ le modèle d'interaction producteur-consommateur recherché dans le RFP.
- TASE.2 fournit un modèle de données hiérarchique fondé sur les abstractions : DataValue, DataSet et VCC. Les VCCs, ou VMDs si il est nécessaire d'adresser des équipements d'aspect general rassemblent les ressources dédiées à un même équipement : DataSetTransferSets, DataSets et DataValues. Les abstractions DataSets regroupent des abstractions DataValues. Les DataValues référencent des points de mesures qui peuvent être simples à très complexes comme nous avons pu le voir préalablement en fonction des attributs associés. On peut étendre au besoin ce modèle à d'autres types de données.
- L'attribut COV (Change of value) qui compte combien de fois une valeur a été modifiée sur le serveur entre deux TransferReports successifs relatifs au même DataValue peut permettre la mise en œuvre de la propriété d'atomicité requise par le niveau de conformité L2.

- Les propriétés de QoS associées aux échanges de données sont spécifiées dans les abstractions DataSetTransferSet du serveur et configurées, dans OpenTAZ, par le client à l'aide d'une interface de gestion précise. Ces propriétés de QoS sont une première contribution pour une conformité partielle de niveau L1.
- Dans OpenTAZ, l'organisation des différents services en 4 interfaces définit une séparation claire entre le mode échange de données opérationnel, la gestion, et l'organisation des interactions entre client et serveur.
- Finalement, et autant que nous pouvons en juger, OpenTAZ est capable de répondre aux besoins des applications du domaine de l'informatique industrielle, représentés au sein de l'OMG par les groupes de travail : the Manufacturing Domain Task Force and the Utility Domain Task Force of OMG.

Pour envisager plus de conformité avec le RFP, c'est-à-dire satisfaire les niveaux L1 et L2, il nous semble nécessaire de construire une véritable mémoire répartie temps réel d'objets qui utiliserait OpenTAZ pour mettre à jour des objets protocolaires dits "ombre" (DataValues ou DataSets) accédés localement par les objets applicatifs suivant le même principe que dans le bus de terrain FIP [LET91]:

- OpenTAZ devrait utiliser un protocole de diffusion intégré à CORBA construit au-dessus de UDP et RTP/RTCP[MEL01] (Real-Time Protocol/Real-Time Control Protocol). RTP fournit un estampillage temporel et des numéros de séquence tandis que RTCP offre des moyens pour gérer une relation RTP entre plusieurs participants. Un protocole à diffusion est obligatoire car certains objets peuvent être répliqués sur des machines différentes, et il faut pouvoir les mettre à jour en même temps chez tous les consommateurs.
- La propriété d'un objet est un aspect abordé dans le RFP, généralement, la propriété d'un objet est liée à celui qui le met à jour (dernier écrivain), ici le producteur. Dans IEC-TASE.2, la propriété ne change pas, mais on pourrait ajouter des méthodes dans l'interface de gestion qui permettrait de le faire. On peut aussi considérer qu'un producteur change dans le temps et s'intéresse à la mise en place d'un service de gestion cohérente de copies multiples réparties entre producteurs et consommateurs. La mise en place d'une mémoire répartie temps réel d'objet est plus simple : l'application écrit dans l'objet "ombre", et la modification est répercutée vers les consommateurs de cet objet en fonction de la sémantique de cohérence choisie ou l'application lit dans l'objet "ombre" la valeur stockée et les attributs associés (date, fraîcheur...). Des sémantiques de cohérence temps réel plus élaborées, comme la cohérence causale temporelle [COR97] peuvent être mises en place.

Ces tâches ne font pas parties de nos prochains travaux. Nous préférons en premier explorer les apports de RT-CORBA [OMG02b] et du canevas offert par l'ordonnancement dynamique temps réel[OMG02a] dans le contexte d'une application réelle.

### 3.3.3.3 Vers plus de temps réel

Un autre type d'extension concerne les aspects temps réel, et plus globalement de QoS temporelle. Considérer la QoS temporelle représente pour ma part un objectif plus global que

le temps réel dans la mesure où les contraintes temporelles sont prises en compte dans tous les composants du système distribué : du réseau jusqu'à l'application en passant par le système d'exploitation et le bus logiciel à objets.

### 3.3.3.3.1 Plate-forme Temps Réel

L'exploration des propriétés temps réel pour les noyaux d'exécution et les systèmes répartis s'est faite à deux occasions : la thèse de C. Lizzi [LIZ99c], et l'étude de plates-formes du commerce pour l'informatique industrielle d'EDF.

L'intégration de propriétés temps réel à tous les niveaux d'un système distribué (réseau, système d'exploitation, bus logiciel) n'est pas chose simple. Les travaux de C. Lizzi ont montré qu'une plate-forme sur mesure, dont les éléments sont intégrés et complétés de façon optimale, donnait d'excellentes performances. C. Lizzi disposait du code source de ChorusOS et avait toute latitude pour le transformer. ChorusOS s'est vu adjoindre un ordonnanceur par échéance de type Earliest Deadline First intégrant l'ordonnancement par priorité classique. Les communications empruntaient un réseau ATM (Asynchronous Transfer Mode) en utilisant au mieux les classes de service, par exemple l'AAL 3/4 servait à véhiculer la priorité des messages permettant un héritage de priorité entre activités (threads) de bout en bout aussi bien en local qu'en distribué. L'ensemble supportait une machine virtuelle Java temps réel minimale construite à partir de Japhar [REI98]. La gestion des tampons mémoire a été optimisée afin de fournir un mécanisme de type zéro-copie de l'espace utilisateur jusqu'au coupleur ATM à travers le noyau et la RT-JVM, même l'interface avec la machine virtuelle; JNI (Java Native Interface) avait été ré-écrite pour cette optimisation. A titre indicatif, le prototype de C. Lizzi permettait une communication de thread Java à thread Java à un débit de 97Mb/s, avec 0,3% de perte de cellules et un délai de 15ms pour un réseau ATM de 155 Mb/s constitué d'un seul commutateur, et des processeurs PowerPC de faible vitesse [LIZ99c]. Dans un contexte différent, l'évaluation de services MMS au-dessus d'un ORB Java sur ATM [SEI99] [BOI01], avec des composants standards, processeurs Intel II de faible vitesse, un ATM à 155Mb/s constitué aussi d'un seul commutateur, Linux, et d'une machine virtuelle Java standard, donnait un débit de 82Mb/s avec un taux de perte variant de 14,1% à 45% et un délai de 90ms. Le prototype de C. Lizzi était largement suffisant pour les applications d'informatique industrielle d'EDF candidates mais beaucoup trop spécifique [LIZ00]. L'architecture de la plate-forme est toujours d'actualité, il faudrait probablement remplacé ATM par MPLS (Multi Protocol Layer Switching).

Pour EDF R&D, pour la branche contrôle-commande en centrale, nous avons essayé de construire une plate-forme équivalente à l'aide de composants commerciaux sur étagère en gardant à l'esprit les résultats obtenus à l'aide du prototype de C. Lizzi [BAC00]. La tâche est très difficile. Le prototype de C. Lizzi permettait un raisonnement complet en échéances très pratique alors que les produits du marché raisonnent en priorité numérique. Les applications modélisent leurs contraintes par des durées ou des dates qui se ramènent à des échéances, il faut alors transformer ces échéances en priorités. Les priorités numériques des différents étages (réseau, noyau, bus logiciel) ne sont pas corrélées et il n'existe aucun moyen standard qui permette de le faire.

Les prochains travaux relatifs à OpenTAZ devraient se concentrer sur RT-CORBA [OMG02a] (chap 24) et le canevas d'ordonnancement dynamique réparti associé [OMG02c], tel que proposé dans l'environnement RapidRMA/RapidSCHED de Tri-Pacific [TRI01]. TAO [TAO02] serait notre ORB cible. Il existe des Linux temps réel que nous pourrions utiliser, mais il faut refaire une évaluation comme nous l'avons fait pour EDF-DER.

Remarquons toutefois que CORBA pose un certain nombre de problèmes. En effet, CORBA se fonde sur TCP, et il est bien connu que TCP n'est pas du tout fait pour le transport de données temps réel. Ses désavantages sont liés à l'introduction de gigue et d'une latence non bornée: gestion des erreurs de transmission, fenêtre glissante, gestion de la congestion (slow start et congestion avoidance). Dans un contexte Internet, UDP conjointement avec RTP/RTCP formeraient la base d'une solution plus efficace et plus intégrable au multicast IP. Plus généralement, les protocoles de transport pour le multimédia pourraient être utilisés pour les communications en usine. Il est à noter que certaines études considèrent CORBA au-dessus d'un bus de terrain [citer les études relevées sur Internet].

#### 3.3.3.3.2 Métaphore du Concert Réparti Virtuel sur réseau haut débit

Actuellement, nous sommes en train de mettre en place les bases d'une expérience sur le temps réel réparti de type multimédia. Un premier stage défrichant les contraintes a eu lieu sous la forme d'expérimentation des protocoles RTP/RTCP pour le transfert de son 3D ambiophonique enregistré à la volé et joué immédiatement dans le cadre de la métaphore de l'orchestre réparti virtuel [BOU02]. Ce travail s'effectue en complète coopération avec l'équipe temps réel de l'IRCAM (Institut de la Recherche en C et Acoustique Musicale) qui est à l'origine du logiciel de traitement de son jMax [IRC02].

Plusieurs musiciens jouent ensemble dans des lieux géographiquement différents reliés par un Internet haut débit. Afin d'achever la métaphore du concert réparti virtuel, le son produit est récupéré, spatialisé, envoyé vers les musiciens et le public. Le public est supposé être lui aussi géographiquement réparti par petits groupes ou isolément, il est supposé être raccordé par un réseau ADSL à l'orchestre virtuel réparti. Outre les contraintes propres au transfert du son (le son étant un flux isochrone, il impose le moins de gigue possible, un taux d'erreur faible, et une latence constante autant que faire se peut), un soin particulier doit être apporté à la latence qui doit être constante pour les musiciens. En effet, les musiciens sont capables de joués en décalé (c'est une contrainte typique des organistes par exemple), mais ce décalage ne doit pas varier dans le temps.

Ce projet est de type logiciel libre. Il permettra la visite de tout l'ensemble des contraintes temps réel en univers réparti, de l'application jusqu'au réseau. La messagerie industrielle OpenTAZ est prévue pour être utilisée dans la partie contrôle-commande et supervision des moyens techniques du concert dont l'opérateur serait l'ingénieur du son.

#### 3.3.3.4 Vers plus de répartition en univers multi-ORB fondée sur l'AOP

Cette partie du projet a pour objectif de développer un service de messagerie industrielle orienté objet indépendamment du bus logiciel à objet cible. Si on fait le bilan des différentes implantations de notre messagerie industrielle, il y a eu plusieurs implantations (en terme de

bus logiciel et de langage), finalement toutes démarrées de zéro : ONC-RPC, différents ORB CORBA, en C, C++ et en Java. L'effort d'implantation demandé n'a pas permis d'explorer Java-RMI, DCOM, ou OPC [CAH01] [OPC02]. Ce que nous souhaiterions maintenant, c'est une fois le langage choisi, Java en l'occurrence, le même service puisse fonctionner au-dessus des ORBs les plus répandus, et ceux des générations à venir. L'AOP (Aspect Oriented Programming) nous semble une bonne approche pour atteindre ce but. L'AOP est une technique d'ingénierie des services systèmes intéressante. Elle paraît être par ailleurs un bon support pour supporter la mise en œuvre du modèle MDA (Model Driven Architecture) de l'OMG [OMG01a].

L'AOP est une approche récente, introduite par Gregor Kiczales de XEROX PARC en 1997 [KIC97]. L'AOP hérite de différents courants d'idées : la séparation des préoccupations issue des méthodes d'analyse et de conception, la réflexivité issue des langages de programmation. Cette approche permet de clairement séparer les différentes fonctionnalités ou bloc fonctionnel d'une application, puis de séparer les blocs fonctionnels des blocs non-fonctionnels (services de trace, de déverminage, mais aussi communication et donc ORB dans notre cas). Une première expérience, AspectTAZ, a été menée avec JAC [PAW01a] [PAW01b] [AOP02], un environnement de développement pour les applications orientées aspects, en cours de réalisation qui réunit le CEDRIC, le LIP6 et le LIFL. JAC offre en standard des aspects Java-RMI, CORBA et SOAP prochainement qui peuvent être utilisées indifféremment et ceci dynamiquement.

AspectTAZ [TEB02a] [TEB02b] est une implantation orientée aspect de IEC-TASE.2 bloc fonctionnel 1 (échanges périodiques) suivant le modèle d'interfaces définit pour OpenTAZ. AspectTAZ bénéficie à la fois des plates-formes CORBA, de JavaRMI et de SOAP sans coût de développement supplémentaire. AspectTAZ pourrait être la base d'un service de négoce d'énergie indépendant des technologies bus logiciels existantes chez les différents partenaires (producteurs, négociants, consommateurs) sur un modèle paire à paire. Ce projet d'application est prévu d'être étudié avec EDF RTE mais ne fait pas l'objet d'un contrat.

Toutefois, du point de vue de nos objectifs de recherche et pour être cohérent avec le paragraphe précédent, il faut étudier l'intégration de propriétés temps réel à JAC. La tâche est conséquente. L'utilisation de JAC avec une JVM temps réel peut être une première étape. Nous avons déjà utilisé PERC [NIL96] [NEW00] lors de l'étude sur le processeur embarqué pour les petites et moyennes unités de production d'énergie [REV01b] et cela s'était avéré concluant: le multi-tâche Java et l'ordonnancement temps réel associé fonctionnaient de façon satisfaisante, le ramasse-miettes était interruptible et incrémental, et dans nos expériences, l'ordonnancement des threads Java n'était pas perturbé. La seconde étape consiste à ajouter une librairie d'aspects temps réel qui permettent à la fois de fournir un CORBA-RT, un Java-RMI-RT, et un SOAP-RT. C'est un véritable défi, mais les extensions temps réel de SOAP et de Java-RMI, même si cela peut paraître paradoxal, pourraient être construites sur la base de la spécification CORBA-RT par transposition. Enfin, on ne peut pas aborder l'univers du temps réel si JAC n'est pas optimisé pour cet objectif, nous avons pu constater le surcoût du à l'environnement d'exécution lors de la mise en œuvre d'AspectTAZ.

### 3.4 Perspectives et Conclusion

L'extension de OO-MMS en QoS-OO-MMS illustre la façon de prendre en compte des contraintes temps réel réparties portant sur les interactions entre objets applicatifs comme par exemple: temps d'exécution d'invocation borné, échéance associée à une invocation, gigue d'invocation... Les paragraphes 3.2 et 3.3 montrent d'ailleurs que le modèle RM-ODP/ReTINA pour l'organisation des interactions entre objets et l'expression des contraintes de QoS temporelle correspond tout à fait à nos besoins[BAC00]. Toutefois, quelle que soit l'architecture de communication à QoS retenue, l'infrastructure d'exécution système doit, elle aussi, être capable de gérer des contraintes temps réel de bout en bout [BAC00]. Pour clore, ce volet d'études liant contraintes temps réel et interactions entre objets, il faudrait explorer l'approche CORBA temps réel [OMG02b], et surtout le canevas pour l'ordonnancement temps réel distribué dynamique [OMG02a].

Le travail qui a mené de la norme IEC TASE.2 à OpenTAZ puis aujourd'hui à AspectTAZ est, comme on a pu le voir, riche en perspectives aussi bien du point de vue des applications que du point de vue de la construction de services systèmes distribués temps réel. Bien sûr, il reste à le relier aux résultats de la partie QoS, ce qui est projeté à travers le projet Concert Réparti avec l'IRCAM.

En l'état de nos résultats, nous disposons de tous les éléments pour construire une base de données temps réel répartie avec des propriétés de cohérence temporelle, tel que nous nous l'étions fixés au cours de l'été 1997 avec Thierry Cornilleau et Frédéric Weis. Cette construction est prévue dans le cadre d'une application d'informatique industrielle de type supervision de processus de production. Deux domaines sont envisagés : la production de pétrole ou de gaz avec Total-Fina-Elf, ou la production d'énergie avec EDF. On s'intéressera à l'utilisation de la programmation par aspects et donc à un univers où les contraintes temporelles ne sont pas strictes.

## 4 Conclusion

### 4.1 Bilans

#### 4.1.1 Bilan Scientifique

Il faut reprendre l'ensemble des travaux, en particulier ceux sur la messagerie industrielle et leur origine. Par rapport aux objectifs initiaux d'offrir une messagerie industrielle sur TCP/IP datant de 1993, on peut conclure que cela anticipait complètement le mouvement industriel qui a démarré en 1998 environ et qui tendait à porter les messageries industrielles propriétaires sur TCP/IP/Ethernet commuté full duplex, ModBus [MOD01] ou PROFINet [POP01] me semblent représentatifs de cet objectif. Notons qu'il existe toujours des travaux liés à l'élaboration d'une solution pour MMS au dessus de TCP/IP [MYO02]. Notre objectification d'une messagerie industrielle correspond à un objectif plus récent des fournisseurs de technologies de l'informatique industrielle, dans ce sens le projet OPC/DX en relation avec les différentes proposition de bus de terrain sur Ethernet et TCP-UDP/IP [CAH01] [OPC02] paraît représentatif de cette tendance récente remontant environ à l'année 2000. Nous pouvons regretter de ne pas avoir soumis nos travaux en 1997, dès qu'ils étaient prêts, à l'OMG, il semble qu'ils avaient une vraie chance de marquer le groupe d'étude du domaine Manufacturing d'après Evan K. Wallace (responsable du groupe de travail "Manufacturing Execution Systems and Machine Control" à l'OMG jusqu'en décembre 2000, Evan K. Wallace travaillait à la Division Manufacturing Systems Integration du NIST).

Un point est à soulever dans le paragraphe précédent qui n'est pas relatif au bilan scientifique. Actuellement, la partie réseau des bus de terrain évolue vers des solutions dont l'élément d'interconnexion est le commutateur Ethernet. Je pense que c'est une erreur fondamentale. Compte tenu des progrès dans la technologie des routeurs, il me semble plus intéressant d'utiliser des routeurs à la place des commutateurs Ethernet :

- Les routeurs savent vraiment gérer de la QoS, même si c'est complexe, avec l'acception de DiffServ, de nombreux mécanismes existent : lissage de trafic, filtrage, ordonnancement des datagrammes ... Ceci implique de "mettre à plat" le réseau d'usine en éliminant la classique pyramide CIM [NUS91]. Avec ce choix, on serait très proche d'une solution construite à partir d'ATM, comme dans la solution CSTI [LIZ99c] ou dans MidART du MERL [SHE00] ... mais avec la famille de protocoles associés à IP.
- Il faut utiliser des fibres optiques, et utiliser IP sur fibre optique comme dans les réseaux de cœur [PUJ02][PUJ01].

Ce genre d'affirmation est facile, et reste à démontrer. Monter une maquette est faisable mais couteux. L'idéal reste une simulation, en utilisant des logiciels comme Opnet []. C'est une forme de retour dans le domaine de l'évaluation de performances, ma discipline d'origine.

L'intégration d'un service de type messagerie industrielle à une base de données temps réel répartie est un besoin récurrent de ce type d'architectures pour les applications d'informatique industrielle, en particulier dans les bus de terrain, l'objectification de l'ensemble dans les solutions techniques des fournisseurs ne devrait plus tarder. Les choix technologiques des fournisseurs ne correspondent pas à CORBA, mais ceci ne remet pas en question les propositions décrites dans ce document. D'ailleurs la communauté des utilisateurs d'informatique industrielle se tourne vers des plates-formes de type web services plus par flexibilité que par soucis de performance.

Les travaux liés à la HDR, ont bien permis de visiter les principales caractéristiques d'une messagerie industrielle moderne.

#### 4.1.2 Bilan Quantitatif

Le bilan de ce projet peut se mesurer suivant plusieurs aspects : publications personnelles<sup>6</sup>, formation, relations industrielles, et enfin, personnel et humain.

##### 4.1.2.1 Bilan des Publications pendant la HDR

Le projet dans sa globalité a généré un certain nombre de travaux, dont les publications sont la partie quantifiable par les autorités d'évaluation. Le tableau ci-dessous donne un aperçu global :

Type de pub.	Rev. Internat	Conf. Internat	Worksh Internat	Conf Nation	Sémin. de rech.	Rapp. Cedric	Rapp. D'Etud.
Quantité	3	14	6	10	8	7	17

Pour plus de détails, le lecteur est renvoyé au CV [GRE02].

##### 4.1.2.2 Bilan en terme de formation

En terme de formation, le projet a permis de former des élèves ingénieurs CNAM, CNAM-IIIE, des étudiants en DEA-DESS-DED, et des Docteurs. Un doctorant est sur le point de terminer sa thèse, sans financement, il est parti en entreprise. Le tableau ci-dessus fait un récapitulatif du volume d'étudiants formés strictement dans le cadre des travaux relatifs à la préparation de cette HDR.

Type de diplôme	Thèse	DESS	1G-CNAM	1G-IIIE	DEA	DED
Quantité	2+1	1	20	2	9	1

A ma connaissance, tous ont trouvé un travail correspondant au niveau de qualification atteint, dans ou proche du domaine de compétences techniques de leur choix pour leur premier emploi après l'obtention du diplôme.

<sup>6</sup> Si on considère l'ensemble du projet, il y a plus de publications. J'encourage les étudiants à soumettre seul des publications. Ca a été le cas de Christophe Lizzi, et d'Erwan Becquet.

### 4.1.2.3 Bilan Industriel

Les coopérations ont été relativement nombreuses par rapport à la surface du projet. Toutefois, elles n'ont peut-être pas mené à un volume de financement aussi important qu'il eut été souhaitable. Les services R&D des entreprises avec lesquelles nous avons développé une coopération industrielle et scientifique étroite sont : Chorus Systems, Aérospatiale, Alcatel-Alsthom Recherche, France Telecom CNET, EDF R&D. Toutefois l'essentiel de ces coopérations était basé sur l'échange, et ces échanges ont été particulièrement fructueux, ils ont d'ailleurs tous menés à des publications. Le tableau ci-dessous donne une vue globale de ces coopérations :

Entreprise	Lieu	Contact Scientifique	Thème
Sun Microsystems (ex Chorus Systèmes), de 1990 à 1997	St Quentin	M. Guillemont, puis M. Rozier, et MI. Ortega	Noyau CHORUS et Mémoire Répartie Partagée
Aérospatiale, de 1993 à 1995	Suresnes	N. Chevassus	Productique
France Telecom équipe ASR, de fin 1993 à 2000	Issy	F. Horn	Temps réel et ORB à QoS
Alcatel ABS en 1996	Colombes	P. Lasserre, E. Thiaville	Mémoire Répartie Partagée et Téléphonie
Dassault Aviation en 1996	St Clouds	Y. Lebiannic, C. Campan	Temps Réel Synchrone pour l'embarqué
Institut Pasteur-SIS, en 1996 et 1997	Paris	L. Bloch, C. Letondal, M. Muller	Parallélisation d'un algorithme de classement de séquences d'ADN sur TreadMarks, application à la recherche sur le SIDA <sup>7</sup>
Alcatel Alsthom recherche de fin 1996 à 1997	Marcousis	S. Chabridon, A. Feray	Etude d'architecture système d'une nouvelle génération d'un superviseur d'applications de contrôle-commande
CSTI, de 1997 à 1999	Lyon	J. Montiel puis P. Gaillard	ATM Temps Réel et Java pour l'Informatique Industrielle
EDF-DER, de fin 1998 à janvier 2001	Chatou	L. Bacon	Contrôle Commande en Centrale de production d'énergie

Malheureusement, nous avons approché de très près le statut de projet pré-industriel avec EDF R&D Chatou au début 2001. Nous étions sur le point de pouvoir réaliser un pilote. Ce pilote devait reposer sur une messagerie industrielle équivalente à TASE.2 sur CORBA pour l'échange de données de production d'énergie électrique provenant des centrales hydrauliques. La raison technique de cet échec semble être liée au refus final côté producteur d'énergie d'utiliser TASE.2 ou une solution fonctionnellement équivalente par les usagers en centrale, préférant des solutions classiques de type fax ou bornier. D'un point de vue contextuel, notre proposition est probablement arrivée à un moment peu opportun où le marché de la fourniture d'énergie devait s'ouvrir.

<sup>7</sup> Müller-Trutwin M., Chaix M-L., Letourneur F., Bégau E., **Beaumont D.**, A. Deslandres, You B., Morgan J., Mathiot C., Barré-Sinoussi F., Saragosti S., Increase of HIV-1 subtype A in Central African Republic. JAIDS (Journal of Acquired Immunodeficiency Syndrome). 21 : 64-171.1999

#### 4.1.2.4 Bilan Personnel

Le bilan personnel porte bien plus sur l'aspect humain que sur l'aspect scientifique. En définitive, les connaissances et les compétences scientifiques acquises à travers les différents projets se révèlent être périssables. Seule perdure l'expérience acquise.

Les travaux liés à la HDR ont été l'occasion de rencontrer de nombreux étudiants aux parcours et projets personnels très divers, avec très majoritairement une grande ouverture d'esprit. Il y a aussi eu des rencontres inattendues parmi les industriels avec qui j'ai coopéré et qui se sont révélées exceptionnelles. Globalement, j'ai eu la chance de rencontrer une grande générosité intellectuelle parmi cet ensemble de personnes. Cette dizaine d'années a été la source d'une expérience personnelle très riche, et passionnante.

La chose qui compte énormément, parce que c'est celle qui laisse une empreinte durable, c'est la rencontre et la découverte des autres. L'âme est une terre étrangère disait Arthur Schnitzer, l'esprit humain est une terre d'aventure que je trouve captivante.

#### 4.2 Prospective et Perspectives

Dans le domaine de l'informatique, la prospective a quelque chose de divinatoire et surtout de très incertain mais c'est un exercice auquel il faut se risquer dans le cadre de la HDR. Le résultat est très personnel et très périssable.

Pour ma part, les perspectives sont guidées par les applications que j'envisage d'adresser. C'est une composante caractéristique de mon travail, mes orientations futures obéissent donc à cette injonction.

Tout d'abord, il faut regarder les domaines en expansion qui vont avoir besoin d'une messagerie industrielle pour le contrôle et la commande distante d'équipements. On est en droit de penser qu'à l'avenir, tout va communiquer avec tout et que les besoins de supervision d'équipements seront colossaux, en particulier hors des entreprises : de la maison (domotique, Internet, télévision numérique, téléphones, PC et ses périphériques partagés, télécommandes de tous genres, périphériques de jeux, équipements ménager... avec des réseaux fixes et des réseaux sans fils, locaux ou longue distance) à l'entreprise (Intranet/Extranet/Internet, réseaux de bâtiment pour le contrôle de la climatisation et des dispositifs de sécurité, portable, PC, PC de poche) en passant par la voiture (réseau embarqué temps réel d'assistance à la conduite, Internet, GPS par satellite ...) et les lieux publics (bornes d'accès Internet ou à des services dans les transports en communs, les musées, les cinémas, les lieux de formation, les centres de soins, immeubles...). A priori, l'univers auquel les solutions de supervision en devenir devront s'adresser sera très diversifié et très hétérogène avec des calculateurs de tous types, mais une prépondérance des processeurs embarqués, et des utilisateurs massivement mobiles, et coopérants. Les solutions devront être extensibles, fiables (haute disponibilité, tolérantes aux pannes, et sécurisées) modulaires avec de nombreux composants logiciels. Ce dernier point étant particulièrement important car on ne peut ré-écrire l'application en bloc à chaque nouvelle génération, les composants évolueront dans le temps et seront mis au point séparément puis intégrés ensuite. Le processus d'intégration revêtira une très importance car il

sera exceptionnel d'avoir la possibilité d'arrêter l'application, voire un sous-ensemble de l'application.

Malgré un bilan scientifique qui me semble positif, ne peut-on pas se poser la question si tous ces efforts n'ont pas été faussés par la nature de l'objectif lui-même. Certes, cet examen est plus facile a posteriori. Reprenons ce qu'est une messagerie industrielle et examinons si le problème n'a pas été finalement abordé par l'angle le moins approprié. Une messagerie industrielle, c'est un ensemble de services qui permettent de contrôler et de commander des équipements industriels distants hétérogènes. On peut remarquer d'ailleurs que le VMD (Virtual Manufacturing Device) joue le rôle d'une machine virtuelle dédiée à l'informatique industrielle. C'est un type de fonction qui existe aussi dans le domaine de la supervision des réseaux. Fondamentalement, ce sont des services de type protocoles réseaux, même si tout se passe en couche 7. Dans le travail proposé dans cette HDR, nous avons remonté ces services réseaux, plutôt non-fonctionnels finalement, à un niveau applicatif (couche 8), même si nous avons exploré certaines pistes qui ont permis de les structurer, en particulier dans le cas de TASE.2. Est-ce qu'il n'aurait pas été plus judicieux, en reprenant le problème à résoudre à son début, d'envisager l'élaboration de solutions par rapport à la gestion de données réparties avec des contraintes de cohérence de type temps réel directement ? Compte tenu de l'hétérogénéité et du besoin d'une machine virtuelle, Java me semble alors l'environnement privilégié de ce type d'applications bien que d'autres plates-formes comme CORBA ou .NET pourraient aussi convenir. Ceci nous mène à des services de type JavaSpace[] de Sun, même si celui-ci devraient être encore enrichis pour satisfaire les contraintes du domaine visé, en particulier pour bien prendre en compte la très grande répartition (par exemple en gérant les JavaSpace suivant un modèle peer-to-peer) et les contraintes temporelles. Ce type de services a visiblement été conçu à l'origine pour les applications embarquées, et conviendrait bien aux univers d'applications décrits ci-dessus.

Il faut enfin se poser le problème de l'ingénierie des applications de supervision et des services associés. C'est une question essentielle. La nature même de ces logiciels pousse à l'intégration de composants fonctionnels et non-fonctionnels développés séparément. Plusieurs approches semblent s'opposer: CCM (CORBA Component Model)[] et l'approche MDA, versus la programmation orientée aspects. Un environnement tel que JAC [AOP02] s'est montré extrêmement prometteur lors des expériences avec TASE.2. Une approche d'assemblage de composants logiciels telle que le propose la société Kelua<sup>8</sup> : Kilim [] ou le produit associé [], est intéressante. Des choix exploratoires récents vont nous amener à étudier les deux dernières approches AOP et assemblage de composants à la Kelua. Toutefois, il ne suffit pas d'adopter une technique adaptée à un problème ou une approche pertinente, il faut que celle-ci soit étayée par un environnement qui permettent de spécifier, concevoir et prouver que l'implantation est conforme. Parmi les trois approches citées plus haut nous n'avons vraiment essayé que l'AOP à travers JAC. Même si les résultats ont été très convaincants, l'AOP est récente et ne semble pas avoir encore acquis toute sa maturité, un

---

<sup>8</sup> A l'attention des rapporteurs : la partie prospective est très récente, et compte tenu du temps dont j'ai pu disposer, il se peut que certaines références soit incertaines. En particulier pour "Kilim" et l'outil associé dont j'ai vu une démonstration faite par F. Horn, peu avant mon entretien avec J-M. Geib.

modèle et un outil de composition d'aspects semble encore à consolider. Toutefois, certains travaux [équipe P. Cointe, travaux de Rémi Douence] ont pour objectif de combler ce vide.

Pour résumer, aujourd'hui voilà les pistes que je préconiserais pour réaliser des services de messagerie industrielle. L'environnement Java pourrait être le langage et la plate-forme privilégiée pour développer des applications de supervision, des services de contrôle et de commande d'équipements distants domestiques aussi bien qu'industriels au sens le plus large du terme. Le service de messagerie industrielle en tant que tel devrait s'évanouir pour laisser place à un service de type JavaSpace avec une gestion des JavaSpace suivant un modèle peer-to-peer indépendant du bus logiciel sous jacent : RMI, SOAP, .NET, CORBA, boîtes aux lettres de type sockets, et des protocoles de communication : famille TCP/IP, protocoles spécifiques, bus de terrain... L'implantation des applications et l'intégration de ces différents services devraient s'appuyer sur une programmation orientée aspects pour bien séparer les parties fonctionnelles des parties non fonctionnelles. L'assemblage des aspects devrait être contrôlé par un environnement ad-hoc vérifiant les propriétés respectives des différents éléments.

On peut rencontrer des services de messagerie industrielle dans un grand nombre d'applications. Les applications multimédia distribuées sont très diverses et m'intéressent. Coté applicatif, elles permettent plus de créativité. Avec le projet Concert Réparti, lié à la thèse de N. Bouillot, s'ajoute une nouvelle voie d'exploration. Ce type d'application est un lieu de nouveaux défis dans lequel se posent des problèmes de systèmes distribués temps réel d'une autre nature. Les contraintes temporelles ne sont plus déterministes mais statistiques. Dans les besoins utilisateurs, il faut intégrer la création artistiques en temps réel. De façon inattendue, cet univers m'oblige à un retour aux sources. La nature des contraintes temps réel rencontrées amènent à la modélisation et à l'évaluation de performances. Probablement que les futurs travaux incluront des analyses de performance soit par simulation, soit par modèle analytique.

### **4.3 Dernières impressions**

Pour conclure un travail de neuf années consécutives, sans congé sabbatique, la préparation d'une HDR représente un travail colossal qui a un coût personnel loin d'être négligeable. On ressent des sentiments très contradictoires : de très grands plaisirs et de profondes déceptions.

Préparer une HDR, c'est comme participer à une coupe du monde des rallyes avec une 2ch et une petite équipe, dont les membres ne sont pas toujours les mêmes mais qui est très chaleureuse. On court contre des équipes professionnelles (INRIA, CNRS ou grandes écoles comme l'ENS ou l'EPFL) et des équipes usine (MERL, Microsoft, Sun), leurs moyens et leur logistique ne sont pas du tout les mêmes. La recherche de sponsor est capitale, d'ailleurs il est très difficile d'avoir les mêmes sponsor que les équipes professionnelles. Quand on se fait lâcher par un sponsor, on sait que pour terminer la prochaine course sur la ligne d'arrivée, il faudra de l'astuce, de l'intuition, de la mobilité dans la course, de la détermination et beaucoup de chance. Surtout qu'on ne se fait jamais lâcher par un sponsor quand ce n'est pas grave, c'est toujours au moment où un résultat était à porté de main. Alors quand on ne termine pas la

course ou quand on se fait doubler sur le poteau, c'est déprimant, on trouve ça tellement injuste. Mais quand on arrive sur la ligne d'arrivée, c'est un vrai bonheur, si en plus, on a pu doubler une équipe professionnelle ... c'est grisant.

## 5 Bibliographie

### 5.1 Références Extérieures au Projet de HDR

#### 5.1.1 Conférences et Ouvrages

- [AKA95] J. Akazan, Z. Mammeri. "Real-Time extensions to MMS". IEEE International Workshop on Factory Communication Systems. WFCS'95. October 4-6 1995. Leyzin. Switzerland.
- [AHA91a] Causal Memory - M. Ahamad, J.E. Burns, P.W. Hutto, G. Neiger - 5<sup>th</sup> Int. Workshop on Dist. Alg., pages 9-30. Springer-Verlag. 1991
- [AHA91b] M. Ahamad, P.W. Hutto, R. John. Implementing and Programming Causal Distributed Shared Memory. Proc. of the 11<sup>th</sup> ICDCS. May 1991.
- [AHA93] Causal Memory: Definitions, implementation, and programming - M. Ahamad, G. Neiger, P. Kohli, J.E. Burns, W. Hutto - Technical Report GIT-CC-93/55, Gatech - 1993
- [AGR94] Mixed Consistency: A Model for Parallel Programming (Extended Abstract). D. Agrawal and M. Choy and H. Leong and A. Singh. ACM. 13<sup>th</sup> PODC. Aug 1994
- [AMZ96] TreadMarks: Shared Memory Computing on Networks of Workstations, C. Amza, A.L. Cox, S. Dwarkadas, P. Keleher, H. Lu, R. Rajamony, W. Yu, and W. Zwaenepoel, IEEE Computer, Vol. 29, No. 2, pp. 18-28. February 1996.
- [ARI98] T. Ariza, F.R. Rubio. "Communicating MMS Events in a Distributed Manufacturing System using CORBA". DCCS'98. 1998.
- [ARI01] T. Ariza, F.J. Fernandez, F.R. Rubio. "Implementing a Virtual Manufacturing Device for MMS/CORBA". ETFA'2001. Juan-les-Pins. France. October 2001.
- [BER87] G. Berry, P. Couronné, G. Gonthier. "Programmation Synchrone des Systèmes Réactifs : le langage Esterel". Techniques et Sciences de l'Informatique. 6(4). 1987.
- [BLA97] G. Blair, J.-B. Stéfani. "Open Distributed Processing and Multimedia". Addison-Wesley. 1997.
- [BOI97] R. Boissier, M. Raddadi, D. Razafindramary, T. Soriano. Object Oriented Approach for the simulation on NC controlled devices. IFAC-IFIP, CIS'97. Belfort. France. May 1997.
- [BRO01] P. Brooks. "Ethernet/IP Industrial Protocol". ETFA'2001. Juan-les-Pins. France. October 2001.
- [CAM01] J. Camerini, A. Chauvet, M. Brill. "Interface for Distributed Automation IDA". ETFA'2001. Juan-les-Pins. France. October 2001.
- [CAD93] R. Cadwallader. "MAP/TOP Migration". EMUG News. February 1993.
- [CHU00] S-T. Chung, O. Gonzales, K. Rmamanritham, C. Shen. "CreMeS: A CORBA Compliant Reflective Memory Based Real-Time Communication Service". Proceedings of IEEE Real Time Systems Symposium. November 2000.
- [CIA98] CIAME. Réseaux de Terrain, Description et Critères de choix. Hermes. 1998.
- [COR90] J.R. Corbin. "The Art of Distributed Applications : Programming Techniques for Remote Procedure Calls". Springer-Verlag SUN Technical Reference Library 1990.
- [DEC01a] J-D. Decotignie. "**titre à retrouver**". Invited Paper. ETFA'2001. Juan-les-Pins. France. October 2001.
- [DEC01b] J-D. Decotignie. "**titre à retrouver**". Invited Paper. Fourth IFAC Conference on Fieldbus Systems and Their Applications. FeT'2001. 14-16 November 2001. Nancy, France.

- [DUM98] B. Dumant, F. Dang Tran, F. Horn, J.-B. Stéfani, "Jonathan: an Open Distributed Processing Environment in Java", in N. Davies, K. Raymond, and J. Seitz, editors, *Middleware'98: IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*, The Lake District, U.K., September 1998.
- [ELL95] J-P. Elloy, P. Molinaro, R. Ricordel. "A temporal extension to the MMS protocol with KerNext tool". *IEEE International Workshop on Factory Communication Systems. WFCS'95*. October 4-6 1995. Leyzin. Switzerland.
- [FEL02] M. Felser, T. Sauter. "The Fieldbus War : History or Short Break Between Battles?". *IEEE International Workshop on Factory Communication Systems. WFCS'2002*. August 28-30 2002. Västerås. Sweden.
- [GON97] O. Gonzales, C. Shen, I. Mizunuma, M. Takegaki. "Implementation and Performance of MidART". *IEEE Workshop on Middleware for Distributed Real-Time Systems and Services*. December 1997.
- [KUS98] K. Kusunoki, I. Imai, H. Ohtani, T. Nakawaji, M. Ohshima, K. Ushijima. "A CORBA-Based Remote Monitoring System for Factory Automation". *Proceedings of the 1<sup>st</sup> IEEE International Symposium on Object Oriented Real-Time Distributed Computing. ISORC'98*. Kyoto Japan. April 20-22. 1998.
- [KIC97] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.M. Loingtier, J. Irwin. "Aspect-oriented programming". In *Proceedings of the 11<sup>th</sup> European Conference on Object-Oriented Programming (ECOOP'97)*, volume 1241 of *Lecture Notes in Computer Science*, pages 220-242. Springer, June 1997.
- [KIC01] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, W. Griswold. "An Overview of AspectJ". *ECOOP 2001. LNCS 2072*. June 2001.
- [LAM78] L. Lamport. "Time, clocks, and the ordering of events in a distributed system". *Communications of the ACM*, 21(7):558-565, July 1978.
- [LIZ99a] C. Lizzi. "Une architecture de traitement temps réel réparti sur réseau ATM pour le système Chorus". *RTS'99*. Paris. Février 1999.
- [LIZ99b] C. Lizzi. "Java Real-Time Distributed Processing over ATM Networks with ChorusOS". *ETFA'99*. Barcelona, Catalogna. Spain. October 1999.
- [MYO02] K-J. Myoung, D-S. Kim, W-H. Kwon. "Implementation of Virtual Factory using MMS Companion Standard". *IEEE International Workshop on Factory Communication Systems. WFCS'2002*. August 28-30 2002. Västerås. Sweden.
- [MAA97] B. Maaref, S. Nasri, P. Sicard. "An Integrated multimedia Communication system for enterprise networks". *Proceedings of the First IEEE Enterprise Networking Mini-Conference. ENM'97*. Montréal. June 1997.
- [MEL01] J-L Mélin. "Qualité de Service sur IP". Eyrolles. Série : Solutions Réseaux. Février 2001.
- [MUL99] Müller-Trutwin M., Chaix M-L., Letourneur F., Bégaud E., Beaumont D., A. Deslandres, You B., Morgan J., Mathiot C., Barré-Sinoussi F., Saragosti S., Increase of HIV-1 subtype A in Central African Republic. *JAIDS (Journal of Acquired Immunodeficiency Syndrome)*. 21 : 64-171.1999
- [NAT86] S. Natkin, E. Gressier. "Etude d'un Configurateur pour le réseau local ARLIC". *Workshop on Computer Performance Evaluation. INRIA. Sophia Antipolis*. April 28-30 1986. France.
- [NAT87] S. Natkin, A. Woog. "ARLIC : An industriel ISO LAN". *EFOC/LAN 1987*. June 1987. Bâle. Suisse.

- [NUS91] H. Nussbaumer. "TéléinformatiqueIV : Messagerie électronique X400, Messagerie Industrielle MMS, Transfert de fichiers FTAM". Presses Polytechniques et Universitaires Romandes. Collection Informatique. 1991.
- [PAP96] W. Papiernik. "Architecture and Design of Modern CNC/Drive Systems". CPIM'96. Int Conf Nurenberg. Germany. May 1996.
- [PAR72] D. Parnas. On the criteria to be used in decomposing systems into modules. Communications of the ACM, 15(12):1053-1058, 1972.
- [PAW01a] R. Pawlak, L. Seinturier, L. Duchien, G. Florin. JAC: A Flexible and Efficient Solution for Aspect-Oriented Programming in Java. Reflection 2001. LNCS 2192. Pages 1-24. September 2001.
- [POP01] M. Popp, P. Wenzel. "PROFInet, Linking Worlds". ETFA'2001. Juan-les-Pins. France. October 2001.
- [PAW01b] R. Pawlak, L. Seinturier, L. Duchien, G. Florin. Dynamic wrappers: handling the composition issue with JAC. TOOLS 2001, Santa-Barbara CA, USA.
- [POL98] A. Polze, D. Plakosh, K. Wallnau. "CORBA in Real-Time Settings : A problem from the Manufacturing Domain". Proceedings of the 1<sup>st</sup> IEEE International Symposium on Object Oriented Real-Time Distributed Computing. ISORC'98. Kyoto Japan. April 20-22. 1998.
- [PUJ00] G. Pujolle, D. Quibech. "Un état de l'art sur la nouvelle génération Internet". DNAC'2000. 2000.
- [PUJ01] G. Pujolle. "La convergence fixe-mobile dans la nouvelle génération Internet, L'Internet ambiant". DNAC'2001. 2001.
- [REI98] P. Reinholdtsen. "Japhar : the Hungry Programmers Open Source Java". Norwegian Unix Users Group. November 1998.
- [ROD90] M.G. Rodd, G.F. Zhao. "RTMMS - An OSI-based Real-Time Messaging System", Journal of Real-Time Systems, Vol 2, 1990.
- [ROS90] M.T. Rose. "The Open Book : a practical perspective on OSI". Prentice Hall. 1990.
- [ROZ92] M. Rozier, V. Abrossimov, F. Armand, I. Boule, M. Gien, M. Guillemont, F. Herrmann, C. Kaiser, S. Langlois, P. Leonard, and W. Neuhauser. Overview of the Chorus distributed operating system. In Proceedings of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures, pages 39--70, April 1992.
- [SAM93] M. Sample. G. Neufeld. "Snacc 1.0 : A High Performance ASN.1 to C/C++ Compiler". University of British Columbia. Canada, February 1993.
- [SCH01] V. Schiffer. "The CIP Family of Fieldbus Protocols and ist newest Member – Ethernet/IP". ETFA'2001. Juan-les-Pins. France. October 2001.
- [SEM99] V. Sempere, J. Mataix, E. Tordera. "Video Transmission on Industrial Processes on MAP Networks using MMS". 7<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation. ETFA'99, Barcelona, Spain, October 1999.
- [SHE00] S. Schen, I. Mizunuma. "RT-CRM:Real-Time Channel Based Reflective Memory". IEEE Transactions on Computers V49.N11. November 2000.
- [SIE96] J. Siegel. "CORBA Fundamentals and Programming". John Wiley 1996.
- [THI96a] Sérialisabilité Causale: un critère intermédiaire entre la sérialisabilité et la cohérence causale. G. Thia-Kime. MPR'96. Bordeaux. Mai 1996.
- [VAL92] Valenzano. Demartini. Ciminiera. "MAP and TOP Communications". Addison Wesley. 1992.

[WU96] Q. Wu, I. Mansour, T. Divoux, "Network resource management for multimedia manufacturing message service" PROMS'96, 3<sup>rd</sup> IEEE International Workshop on Protocols for Multimedia Systems, Madrid. 15-18 octobre 1996, A. Azcorra Editor, ISBN 84-605-5621-2, 1996.

### 5.1.2 Normes et Standards

[EPR98a] EPRI. Utility Communications Architecture Version 2.0, Introduction to UCA Version 2.0, Editorial Draft 1.0. September, 1998.

[EPR98b] EPRI. Utility Communications Architecture Version 2.0, Common Application Service Models (CASM) and Mapping to MMS, Editorial Draft 1.6. Prepared by KEMA Consulting. September, 1998.

[EPR98c] EPRI. Utility Communications Architecture Version 2.0, Generic Object Models for Substation & Feeder Equipment (GOMSFE), Version 0.81. October, 1998.

[IEC96a] IEC, Utility Communications Specification Working Group. TASE.2 Services and Protocol, Version 1996-08. IEC870-6-503. ICCP Inter-Control Centre Communications Protocol Version 6.1. August 1996.

[IEC96b] IEC, Utility Communications Specification Working Group. TASE.2 Object Models, Version 1996-08. IEC870-6-802. ICCP Inter-Control Centre Communications Protocol Version 6.1. August 1996.

[IEC97] IEC, Utility Communications Specification Working Group. TASE.2 Profiles, Version 1996-11. IEC870-6-702. ICCP Inter-Control Centre Communications Protocol Version 6.1. February 1997.

[ISO90] ISO 9506-1. "Industrial Automation Systems - Manufacturing Message Specification - Part 1: Service Definition". 1990.

[ISO91a] ISO 9506-2. "Industrial Automation Systems - Manufacturing Message Specification - Part 2: Protocol Specification". 1991.

[ISO91b] ISO 9506-4. "Industrial Automation Systems - Manufacturing Message Specification - Part 4: Companion Standard for Numerical Control". 1991.

[OMG00] OMG. "Smart Transducers Interface". RFP. orbos/2000-12-13. December 2000.

[OMG01a] OMG Architecture Board MDA Drafting Team. "Model Driven Architecture (MDA)". Document number ormsc/2001-07-01. July 9, 2001. <ftp://ftp.omg.org/pub/docs/ormsc/01-07-01.pdf>.

[OMG01b] OMG. Real Time CORBA 2.0 : Dynamic Scheduling Specification. <http://cgi.omg.org/docs/ptc/01-08-34.pdf>. September 2001.

[OMG01c] OMG. Data Distribution Service for Real-Time Systems RFP. V 1.0. orbos/2001-10-01. October 2001.

[OMG02a] OMG. The Common Object Request Broker: Architecture and Specification. Revision 3.0. <http://cgi.omg.org/docs/formal/02-06-01.pdf>. July 2002.

[OMG02b] OMG. Minimum CORBA Specification, Version 1. <http://cgi.omg.org/docs/formal/02-08-01.pdf>. August 2002.

[OMG02c] OMG. Real Time CORBA Specification, Version 1.1. <http://cgi.omg.org/docs/formal/02-08-02.pdf>. August 2002.

[POU97] Y. Pouffary, A. Young. "ISO Transport Service on top of TCP (ITOT)". RFC2126. March 1997

[ROS87] M. T. Rose, D.E. Cass. "ISO Transport Services on top of the TCP Version 3". RFC 1006. May 1987.

[WAL97] S. Waldbusser. "Remote Network Monitoring Management Information Base Version 2 using SMIV2". RFC2021. 1997.

### 5.1.3 Rapports de Projets Européens, Documentations Techniques et Rapports de Recherche

[BOY92] A causal distributed shared memory based on external pagers. F. Boyer. Rapport technique Bull-IMAG n° 16-92, 22 pages, 1992

[BAC98x] L. Bacon, E. Becquet. "SETR : systèmes d'exploitation temps réel - PSOS – architecture". HP-32/98/097/A. EDF-DER. 1998.

[BAC99x] L. Bacon, E. Becquet. SETR : systèmes d'exploitation temps réel - PSOS - tests. HP-32/99/024/A. EDF-DER. 1999.

[BECyy] E. Becquet, L. Bacon "évaluation de VxWorks"

[CAH01] Cahners Europe. "OPC Foundation demonstrates OPC XML and sets its sights on a new 'DX' standard for Ethernet-based control systems". Control Engineering Europe. October 2001.

[CCE94] CCE-CNMA Consortium. "The CIME Computing Environment - A platform for the creation and execution of industrial applications". Esprit Project 7096. 7096.94.08/F2.PD. 21 April 1994.

[CHO96a] Chorus Systems. "Chorus/COOL-ORB Programmer's Guide". CS/TR-96-2.1. 1996.

[CHO96b] Chorus Systems. "CHORUS/ClassiX r3, Technical Overview". CS/TR-96-119.9. 1996.

[DECxx] P. Decogné, L. Bacon. "évaluation de LynxOS".

[GRE86] E. Gressier. "Contrôle Réseau d'ARLIC". Retrouver ref . SEMA GROUP. 1986

[JOH94] Evaluation of causal distributed shared memory for data-race-free programs - R. John, M. Ahamad - Technical Report GIT-CC-34, Gatech - 1994

[KEM96] KEMA-ECC. ICCP User Guide. Final Draft. Mineapolis. October 8 1996.

[LET91] P. Leterrier, J-P. Thomesse. "Le bus de terrain FIP". Personnel Communication. 1991.

[LOG ] C. Logé, L. Bacon. "DCOM et le temps réel". HP32/00/099/A. EDF-DER. 2000.

[MER98] A. Mermod, G. Genilloud. "MMS over CORBA". Personnel communication. February 1998.

[NIN92] Nine Tiles Computer Systems Limited. "MMS : The Language for the Integration of Manufacturing Devices". 1992.

[RAY92] Causality oriented shared memory for distributed systems - M. Raynal, M. Mizuno, M. Neilsen - INRIA, RR-656, 8 pages - 1992

[RAY95a] From Causal Consistency to Sequential Consistency in Shared Memory Systems. M. Raynal, C. Schiper. Rapport de Recherche INRIA RR-2557. Mai 1995.

[RAY95b] A suite of Formal Definitions for Consistency Criteria in Distributed Shared Memories. M. Raynal. A. Schiper. PI n°968. IRISA. Novembre 1995.

[REE02] Reed Business Information. Control Engineering Europe. September 2002.

[SEM94] SEMA GROUP - ARLIC. "MMS Communication profile over TCP/IP". Fiche Produit. 1994.

[TIN95] TINA-C. "Overall Concepts and Principles of TINA". Version 1.0. TINA-C Deliverable. February 1995.

[VME96] VME International Corporation. "Reflective Memory DMA Performance White Paper". Document No 840-000005-000. July 1996.

#### 5.1.4 Thèses

[JOH95] Implementing and Programming Weakly Consistent Memories . R. John. PhD Thesis. Georgia Institute of Technology. Mars 1995. GIT-CC-95-12.

[GRE87] E. Gressier. : "La Modélisation au Moyen des Réseaux de Petri Stochastiques : Application aux Systèmes Informatiques fortement synchronisés". Thèse de Doctorat du CNAM. Octobre 1987.

[ORT92] M-I. Ortega. "La mémoire Virtuelle Partagée Répartie au sein du Système Chorus". Thèse de Doctorat de l'Université Paris 7. Décembre 1992.

[PUA93] I. Puaut. "Gestion d'objets actifs dans les systèmes distribués: problématique et mise en œuvre". Thèse de Doctorat de l'Université de Rennes 1. Janvier 1993.

[WEI96] F. Weis. "Une approche pour l'Intégration d'Applications MMS". Thèse de Doctorat du Conservatoire National des Arts et Métiers. Paris. 18 Septembre 1996.

#### 5.1.5 Documentations issues du Web

[ALL95] A. Alles, ATM Internetworking, Proceedings of Engineering InterOp, Mars 1995. <http://cell-relay.indiana.edu/cell-relay/docs/cisco.html>.

[ALM97] W. Almesberger, ATM on Linux - the 3<sup>rd</sup> year, 4<sup>th</sup> International Linux Kongress, <http://lrcwww.epfl.ch/linux-atm/>, Mars 1997.

[AOP02] JAC. <http://jac.aopsys.com>. April 2002.

[ATM96a] ATM Forum Technical Committee, ATM P-NNI Specification Version 1.0, Mars 1996. <ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000.pdf>.

[ATM96b] ATM Forum Technical Committee, ATM UNI Signalling Specification Version 4.0, <ftp://ftp.atmforum.com/pub/approved-specs/af-sig-0061.000.pdf>.

[FIE02] Fieldbus Foundation. "Foundation Fieldbus Technical Overview". <http://www.fieldbus.org/>. October 2002.

[IAO02] IAONA. Ethernet in Automation. <http://www.iaona-eu.com/>. April 2002.

[IBU00] IButton. "Introducing TINI: Tiny InterNet Interface" <http://www.ibutton.com/TINI/index.html>, September 2000.

[IRC02] IRCAM. jMax home page. <http://www.ircam.fr/equipes/temps-reel/jmax>. 2002.

[LAU98] M. Laukien, U. Seimet. "ORBacus version 3.1.2". Object-Oriented Concepts Inc, 1998. <http://www.ooc.com>

[MIC02] MICO. <http://www.mico.org/>. April 2002.

[NEW00] Newmonics Inc.: <http://www.newmonics.com/>, October 6th 2000.

[NIL96] K. Nilsen, Issues in the Design and Implementation of Real-Time Java, Technical Report NewMonics Inc, <http://www.newmonics.com>.

[MOD01] Modbus/TCP Homepage: Automation Business. <http://www.modicon.com/openmbus/>. May 2001.

[OBJ01] Object Web Project: <http://www.objectweb.org/>. March 31st.2001.

[OOC99]. OOC. Object Oriented Concepts. ORBacus presentation. <http://www.ooc.com/ob.1999>.

[OPC02] OLE for Process Control foundation. <http://www.opcfoundation.org>. October 2002.

[OPE02] OpenTAZ project. <http://savannah.nongnu.org/projects/opentaz>. October 2002.

[SMI97] Smile. "gdidl 1.8 ASN.1 to IDL translator". <http://www.smile.fr/news.html>

[SYS02] SYSTRAN. "SCRAMNet+ Shared Memory – Speed, Determinism, and Flexibility For Distributed Real-Time Systems". <http://www.systran.com/>. October 2002.

[TAO02] TAO. <http://www.cs.wustl.edu/~schmidt/Tao.html>. April 2002.

[TRI01] Tri-Pacific. <http://www.tripac.com/>. March 2001.

[ZEN02] ZEN. Real-Time CORBA with the ZEN ORB. DOC Group. <http://www.zen.uci.edu/>. April 2002.

## 5.2 Bibliographie personnelle relevant de la HDR

Cette bibliographie est issue de

[GRE02] E. Gressier-Soudan. Curriculum Vitae Maître de conférence- Habilitation à Diriger des Recherches. Novembre 2002.

### 5.2.1 Revues Internationales

[BOI01] Boissier R., Gressier-Soudan E., Laurent A., Seinturier L., "Enhancing Numerical Controllers using MMS Concepts and a CORBA based software bus". International Journal of Computer-Integrated Manufacturing (IJCIM), V14 N6, 2001. Taylor and Francis Editor.

[COR96] T. Cornilleau, E. Gressier-Soudan. "A combined Consistency Approach: Sequential&Causal-Consistency". ACM Operating System Review.V30. N4. p33-44. October 1996.

[GRE99a] Gressier-Soudan E., Epivent M, Laurent A., Boissier R., Razafindramary D., Raddadi M., "Component oriented control architecture, the COCA project" Special Issue on Manufacturing, Microprocessors and Microsystems Journal - V23N2 September 1999, p95-102, Elsevier Science.

### 5.2.2 Revue Nationale Electronique

[GRE01b] E. Gressier-Soudan, C. Logé, L. Bacon. "Construction de plates-formes à objets réparties temps réel pour les applications d'information industrielle ". RERIR, numéro 11. Mars 2001. <http://rerir.univ-pau.fr/rerir2.html>

### 5.2.3 Chapitre de Livre

[RAZ96] D. Razafindramary. M. Fofana. E. Gressier-Soudan. M. Raddadi. R. Boissier. R. Ponsonnet. "Architecture Réceptacle et services pour l'intégration CFAO-Usinage dans les petites entreprises". Integrated Design and Manufacturing in Mechanical Engineering. Avril 1996. Nantes. France. published in "selected papers of the IDMME'96 Conference. KLUWER Publ., P. Chedmail, J-C. Bocquet, D. Dornfeld Editors.

### 5.2.4 Conférences d'Audience Internationale

#### 5.2.4.1 Session normale

[BAC00] L. Bacon, E. Becquet, E. Gressier-Soudan, C. Lizzi, C. Logé, L. Réveilleau. "Provisioning QoS in Real-Time Distributed Object Architectures for Power Plant Control Applications". Proceedings of the 2<sup>nd</sup> International Symposium on Distributed Objects and Applications, DOA'00. OMG, Antwerp, Belgium, September 2000.

[BEC01b] E. Becquet, E. Gressier-Soudan, E. Legrand, G. Hellack. " Experimenting a Real-Time Industrial Messaging Service for a Power Utility: TASE.2 over the real-time micro-kernel pSOS+". 8<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation. ETFA'2001, Nice, France, October 2001.

[BEC01c] E. Becquet, M. Abdallah, E. Gressier-Soudan, F. Horn, L. Bacon. "Object Oriented Timed Messaging Service for Industrial Ethernet : a Fieldbus like Architecture for Power

Plant Control and Factory Automation". Fourth IFAC Conference on Fieldbus Systems and Their Applications. FeT'2001. 14-16 November 2001. Nancy, France.

[COR95] T. Cornilleau, E. Gressier, M-I. Ortega. "A Multiconsistency Memory Protocol Test Environment using Chorus". European Research Seminar on Advanced Distributed Systems. April 1995. Alpe d'Huez. France.

[GRE95] E. Gressier, M. Lefevbre, S. Natkin. "TCP/IP Manufacturing Applications: an experiment with MMS over RPC ". ULPA'95. Sidney 1995. <ftp://ftp.cs.su.oz.au/bob/ULPAA/101-GRE.ps.gz>

[GRE98] Gressier-Soudan E., Epivent M, Laurent A., Boissier R., D. Razafindramary, M. Raddadi. "Component oriented control architecture, the COCA project". Workshop on European Scientific Industrial Collaboration on promoting Advanced Technologies in Manufacturing. WESIC'98. Gironne. Spain. June 98.

[LIZ00] C. Lizzi, L. Bacon, E. Becquet, E. Gressier-Soudan. "Prototyping QoS based Architecture for Power Plant Control Applications". IEEE International Workshop on Factory Communication Systems 2000. Porto. Portugal. August. 2000.

[LIZ98a] Lizzi C., Montiel J., E. Gressier-Soudan. "A Real-Time Communication Service for ATM-based Distributed Systems". IEEE International Conference on ATM. ICATM'98. Colmar France.

[LIZ98b] Lizzi C., Gressier-Soudan E., "A real-Time IPC Service over ATM Networks for the Chorus Distributed System". 24<sup>th</sup> Euromicro Conference. Vasteras. Sweden. August 1998.

[SEI99] Seinturier, L., Laurent, A., Dumant, B., Gressier-Soudan, E., Horn, F., "A framework for Real-Time Communication Based Object Oriented Industrial Messaging Services". 7<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation. ETFA'99, Barcelona, Spain, October 1999.

[TEB02] L. Teboul, R. Pawlak, L. Seinturier, E. Gressier-Soudan, E. Becquet. "AspectTAZ : a new approach based on Aspect Oriented Programming for Object Oriented Industrial Messaging Services design". IEEE International Workshop on Factory Communication Systems 2002. Vasteras. Sweden. August. 2002.

#### **5.2.4.2 Session Work In Progress**

[REV01b] L. Réveilleau, E. Becquet, E. Gressier-Soudan, JM. Douin. F. Horn, L. Bacon., "Towards a RT-Java Based Embedded Remote Monitoring Tool for Small and Medium Power Plant Units". Work in Progress Session. 8<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation. ETFA'2001, Nice, France, October 2001.

[THI96b] V. Thiebold, T. Cornilleau, E. Gressier-Soudan. M.I. Ortega. "Implementing a Causal Distributed Shared Memory on CHORUS". PDCS'96. Work in Progress Session. Dijon. September 1996.

#### **5.2.5 Workshops à Audience Internationale**

[BOI98] Boissier R., Epivent M., Gressier-Soudan E., Horn F., Laurent A., Razafindramary D., "Providing Real-Time Object-Oriented Industrial Messaging Services". Workshop OORTS: Object Oriented Technology and Real-Time Systems. Actes ECOOP'98 Springer Verlag. July 1998. Brussels. Belgium.

[COR97c] Cornilleau T, Gressier-Soudan E. "RT-Objects based on Temporal Causal Consistency :A new approach for fieldbus systems". ECOOP'97. Workshop : Real Time Objects. Jyväskylä. Finland. June 1997.

[GRE00a] E. Gressier-Soudan. "Prototyping a CORBA based MMS -Industrial Communications with CORBA-". OMG Technical Meeting. Burlingame, California USA. September 10-15 2000. <ftp://ftp.omg.org/pub/docs/mfg/00-09-16.pdf>.

[GRE01a] E. Gressier-Soudan, E. Becquet. "Real-Time CORBA MMS for Embedded Systems". OMG's First Workshop on Embedded Object-based Systems. January 17-19, 2001. Santa Clara, CA, USA.

[GRE96] E. Gressier-Soudan. "MMS over chorus/cool, an omg/corba approach for Manufacturing Applications". Chorus JAM SESSION'96. Juin 1996. Paris.

[GUY97b] Guyonnet G., Gressier-Soudan E., Weis F. "COOL-MMS: a CORBA approach to ISO-MMS". ECOOP'97. Workshop : CORBA: Implementation, Use and Evaluation. Jyväskylä. Finland. June 1997.

### 5.2.6 Conférences à Audience Nationale

[BEC01a] E. Becquet, P. Decogné, E. Gressier-Soudan, L. Bacon. "Evaluation de Systèmes d'Exploitation Temps Réel pour les Applications d'Informatique Industrielle d'EDF". Real Time Systems 2001. Paris. Mars 2001.

[CAL96] G. Calliet, T. Cornilleau, E. Gressier, C. Toinard. "Mémoire partagée répartie sur groupes de diffusion ordonnée". MPR'96. Mai 96. Bordeaux.

[CHA98] S. Chabridon, V. Delebarre, A. Feray, E. Gressier-Soudan, F. Raviart. "Services système pour la supervision de processus industriel : Traitement d'une Base de Données Temps Réel par modèle synchrone et accès distant à travers CORBA". Real Time Systems 98. Paris. Janvier 98.

[COR96a] Cornilleau, V. Gal, E. Gressier, V. Thiébold. "Cohérence causale uniforme à gestion par versions de pages". Renpar8. Mai 96. Bordeaux.

[COR97b] Cornilleau, E. Gressier-Soudan. F. Horn. C. Lizzi. "Cohérence Causale Temporelle". Real Time Systems 97. Paris. Janvier 1997. meilleur article de la conférence scientifique RTS'97

[GAL95b] V. Gal, T. Cornilleau, E. Gressier. "Modélisation en Lotos d'une Mémoire Virtuelle Répartie à Cohérence Causale". Journées Jeunes chercheurs. Rennes. Octobre 1995.

[GRE00b] E. Gressier-Soudan, C. Logé, L. Bacon. "QoS<sub>8</sub><sup>temporelle</sup> (Application) = CoS(Système d'Exploitation) + CoS<sub>6-1</sub>(Communications) + CoS<sub>7</sub>(Bus Logiciel)". NOTERE'2000. Paris. Décembre 2000.

[GRE96] E. Gressier. M. Lefebvre. M. Rozier. "Protocole de messagerie MMS sur CHORUS: différentes approches". Real Time Systems 96. Paris. France. Janvier 1996.

[GUY97a] G. Guyonnet, E. Gressier-Soudan. JY. Simiand. F. Weis. "Une approche CORBA pour MMS : COOL-MMS". Real Time Systems 97. Paris. France. Janvier 1997.

[LEF95a] M. Lefebvre, E. Gressier, S. Natkin. "MMS sur TCP/IP : une nouvelle solution pour l'échange de données en informatique de production". Real Time Systems 95. Paris. France. Janvier 1995.

### 5.2.7 Rapports de Recherche CEDRIC

[GRE99b] E. Gressier-Soudan, L. Seinturier, "Object Oriented Messaging Services : Which Approach to address Real-Time constraints". Rapport de Recherche CEDRIC 99-16. 1999.

[COR96b] T. Cornilleau, V. Gal, E. Gressier, V. Thiébold. "Mémoire à cohérence causale uniforme: estampillage par horloge vectorielle processeur ou par numéros de version"- Rapport de recherche CEDRIC 96-08. 1996.

[BOR96] S. Bortzmeyer, T. Cornilleau, G. Florin, E. Gressier. "Web Caching and Causal Distributed Shared Memory". Rapport de recherche CEDRIC 96-04. 1996.

[COR95] T. Cornilleau, E. Gressier, V. Gal. "Cohérences Causales et Mémoire Partagée Répartie". Rapport de recherche CEDRIC 95-01. 1995.

### 5.2.8 Rapports d'Etude sur contrats avec des Industriels

[BAC98a] L. Bacon, E. Gressier-Soudan. "Etude des contraintes utilisateurs pour les systèmes d'exploitation temps réel– Questionnaire". EDF-DER. HP-32/98/056/A. 1998.

[BAC98b] L. Bacon, E. Becquet, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - Etude des contraintes utilisateurs - Critères d'évaluation". HP-32/98/078/A. 1998.

[BAC98c] L. Bacon, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - préconisation d'utilisation et normes". EDF-DER. HP-32/98/095/A. 1998.

[BAC98d] L. Bacon, E. Becquet, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - Conception des Tests de Systèmes d'Exploitation du Commerce pour les Applications d'Informatique Industrielle d'EDF- Conception Générale". EDF-DER. HP-32/98/0XX/A. 1998.

[BAC99a] L. Bacon, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - Etude des contraintes utilisateurs - Entretien DER/REME - Application PSAD". EDF-DER. HP-32/99/012/A. 1999.

[BAC99b] L. Bacon, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - Etude des contraintes utilisateurs - Entretien SEMA - Application PSAD". EDF-DER. HP-32/99/013/A. 1999.

[BAC99c] L. Bacon, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - Etude des contraintes utilisateurs - Entretien SEMA - Application CCN4". EDF-DER. HP-32/99/014/A. 1999.

[BAC99d] L. Bacon, E. Gressier-Soudan. "SETR : systèmes d'exploitation temps réel - Etude des contraintes utilisateurs – Synthèse". EDF-DER. HP-32/99/052/A. 1999.

[BAC99e] E. Gressier-Soudan, E. Becquet, E. Legrand, O. Mallet, L. Bacon. "SETR: Systèmes d'exploitation temps réel : Portage d'une souche TASE.2 sur le micro-noyau temps réel pSOS+". EDF-DER. HP-32/00/057/A. 2000.

[BEC00] E. Becquet, E. Gressier-Soudan, L. Bacon. "CORBA et Temps Réel". EDF-DER. HP32/00/098/A. 2000.

[CHAB97a] S. Chabridon, E. Gressier. "P3200NG Real-Time Database - Comparison of Real-Time Operating Systems". Alcatel Alsthom Recherche. UAR/C/97/0106, 31/03/97.

[CHAB97b] S. Chabridon, V. Delebarre, E. Gressier. "P3200NG Software Specification Document", UAR/C/97/0232. 31/07/97.

[CHAB97c] S. Chabridon, V. Delebarre, E. Gressier. "P3200NG Real-Time Database - Software Preliminary Design Document". Alcatel Alsthom Recherche. UAR/C/97/0277. 31/10/97.

[CHE93] N. Chevassus. P. Curnier. R. Gaches. E. Gressier. S. Natkin. C. Toinard. "Pilot Requirement Definition - Aérospatiale Demonstration Facility". Esprit Project 7096. CIME Computing Environment Integrating a Communication Network for Manufacturing Applications (CCE-CNMA). Deliverable 3. Workpackage 7. July 1993. Editor: AEROSPATIALE. CCE-CNMA Consortium. Private Access Document.

[ITC92] ITCIM Consortium. ACERLI. "Integration Testing for Computer Integrated Manufacturing : Test Methodology". Esprit Project N°6860. Esprit 3. WP200. Deliverable 210.1. December 1992.

[ITC93] ITCIM Consortium. Semagroup Spain. "Users Requirements". Esprit Project N°6860. Esprit 3. WP 200. Deliverable D220.1. June 1993.

[REV00] L. Réveilleau, C. Lizzi, E. Gressier-Soudan, L. Bacon. "Java et le Temps Réel". EDF-DER. HP32/00/087/A. EDF-DER. 2000.

### 5.3 Travaux d'étudiants liés à la HDR

Tous les documents indiqués ci-dessous correspondent à des travaux soutenus.

#### 5.3.1 Thèses

[COR97a] T. Cornilleau. "Etude des Cohérences Mémoire Uniformes - Cohérence Causale : mise en oeuvre et Extensions". Thèse de Doctorat du Conservatoire National des Arts et Métiers. Paris. 7 Janvier 1997.

[LIZ99c] C. Lizzi. "Conception d'un Système Distribué Temps Réel Fondé sur ATM". Thèse de Doctorat du Conservatoire National des Arts et Métiers. Paris. 14 Décembre 1999.

#### 5.3.2 DEA

[BAI98] A. Bailly. " An Assessment of Rollback-Dependency Trackability in a Distributed Shared Memory Environment". Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst. 1998.

[BEC97] E. Becquet. "Réalisation d'un mappeur de cohérence séquentielle sous CHORUS r3.6". Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst. 1997.

[BOU02] N. Bouillot. "Services de transport de flux audio 3D pour le concert virtuel réparti". Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst. 2002.

[CAL95a] G. Calliet. " Mémoire Virtuelle Répartie Cohérente sur Groupes de Diffusion : Une étude préliminaire à l'implantation dans Chorus". Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst. 1995.

[COR92] T. Cornilleau. Modélisation de la Mémoire Virtuelle Répartie du Système d'Exploitation Chorus. Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst. 1992.

[LIZ95] C. Lizzi . Contribution au Projet Arcade : un driver FDDI pour CL-Chorus. Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst. 1995.

[TEB02] L. Teboul. "AspectTAZ : implantation d'une messagerie industrielle TASE.2 en utilisant la programmation orientée Aspect". Rapport de stage de DEA Systèmes Paris 6-Cnam-Enst et Mémoire d'Ingénieur CNAM-III. 2002.

#### 5.3.3 Mémoires d'ingénieurs

[ARA96] A. Arazo. " Etude et Implantation d'un service de liaisons temps réel dans une plateforme à objets répartis". Mémoire d'ingénieur Cnam. Paris.1996.

[BEA97] D. Beaumont."VeryFastDNAmI : FastDNAmI parallélisé sur TreadMarks". Mémoire d'ingénieur Cnam. Paris. 1997.

[BOL97] T. Bollereau. "Conception et Réalisation d'une mémoire virtuelle répartie partagée à cohérence séquentielle pour la migration d'objets dans le système CHORUS". Mémoire d'ingénieur Cnam. Paris.1997.

[BON96] P. Bonnel. "Conception et Réalisation d'un service de projection d'Objets sur CHORUS". Mémoire d'ingénieur Cnam. Paris.1996.

[BOU97] J-Y. Bourles. "TreadMarks sur CHORUS mise en oeuvre avec la technologie mappeur". Mémoire d'ingénieur Cnam. Paris.1997.

[CAL95b] G. Calliet. "Mémoire Virtuelle Répartie Cohérente sur Groupes de Diffusion : Propositions d'Algorithmes". Mémoire d'ingénieur Cnam. Paris.1995.

[DEC99] P. Decogné. "Expérimentation de Systèmes d'Exploitation Temps Réel à sémantique Unix, de RT-Linux à LynxOS". Mémoire d'ingénieur Cnam. Paris. 1999.

- [GAL95a] V. Gal. "Spécification en Lotos d'un algorithme de gestion d'une Mémoire Partagée Répartie à Cohérence Causale". Mémoire d'ingénieur Cnam. Paris.1995.
- [GAU94] O. Gaultier, O. Métais. "Mise en Place d'une Plate-forme Chorus, Conception et Implantation d'un Ordonnanceur à échéances au sein du micro-noyau Chorus". Mémoires d'ingénieur Cnam. Paris. 1994.
- [GUY02] G.Guyonnet, "Objectification d'un protocole de messagerie OSI : Implantation de MMS sur COOL(CORBA)/CHORUS". Mémoire d'ingénieur CNAM. Septembre 2002.
- [LAU99] A. Laurent, "Messagerie industrielle portable orientée objet : Application au pilotage distant d'une machine outil à commande numérique". Mémoire d'ingénieur Cnam. Paris. 1999.
- [LEF95b] M. Lefebvre, "MMS sur TCP/IP : une solution qui utilise l'appel de procédure à distance". Mémoire d'ingénieur Cnam. Paris. Janvier 1995.
- [MOL99] O. Molet. "Support d'objets Cohérents en mémoire répartie partagée : Ingénierie sous Chorus". Mémoire d'ingénieur Cnam. Paris.1999.
- [REV01a] L. Réveilleau. "Prototypage d'une station légère pour la supervision à distance d'unités de production d'énergie petites ou moyennes". Mémoire d'ingénieur Cnam. Paris. 2001.
- [RIZ00] A. Rizkalla. "Programmation de la QoS dans les réseaux ATM et Internet". Mémoire d'ingénieur Cnam. Paris. 2000.
- [ROS96] L. Rosenfeld. "Mémoire Répartie dans un Autocommutateur Alcatel A4400". Mémoire d'ingénieur Cnam. Paris.1996.
- [SAU99] C. Sautereau. "TreadMarks au-dessus du micro-noyau Chorus sous forme de bibliothèque". Mémoire d'ingénieur Cnam. Paris. 1999.
- [SIM96] J-Y. Simiand. "Intégration de la Messagerie Industrielle MMS au système chorus". Mémoire d'ingénieur Cnam-IIIE. Paris. Septembre 1996.
- [SIN96]. F. Singhoff. "Mise en oeuvre du Modèle Saturne sur CHORUS/COOL". Mémoire d'ingénieur Cnam. Paris.1996.
- [THI97] V. Thiebold. "Mémoire Répartie Partagée sur CHORUS : de l'implantation de la cohérence causale vers un mappeur générique". Mémoire d'ingénieur Cnam. Paris.1997.
- [TUA87] Tu Anh. "Techniques de l'Intelligence Artificielle appliquées à la Supervision du réseau local industriel ARLIC". Mémoire d'ingénieur Cnam. Paris.1987.

#### **5.3.4 Stages de Laboratoire**

- [EPI98] Réalisation d'un driver sous Linux pour carte de commande numérique. Rapport de stage au GRPI. CEDRIC/CNAM. Paris. Septembre 1998.
- [GUI00] B. Guillon. Web based lightweight HMI for Power Plant Control. DEST final report. September 2000.