

**XML – Le futur de l'interopérabilité – Une  
solution appliquée aux Systèmes d'information  
de commandement et contrôle de la Force  
terrestre**

**LFTSP**



**PEMTFT**

**21 mai 2002  
Major Normand Rochon  
Capitaine Roger Otis**

**Le Collège militaire royal du Canada**

**Département des Sciences militaires appliquées**

**Programme d'état-major technique des forces terrestres VII**

**Projet de recherche en équipe**

Langage de balisage extensible (XML) – L'avenir de l'interopérabilité -  
– Une solution pour les systèmes d'information des forces terrestres.

Le major Normand Rochon et le capitaine Roger Otis

Abstrait

La numérisation du champ de bataille est inévitable. Elle a déjà commencé par l'introduction graduelle de systèmes d'armes numérisées capables de s'acquitter de leurs fonctions de façon indépendante dans leur domaine d'action. La prépondérance de cette situation amène toutefois de nouvelles préoccupations générées par la répétition de la capture des mêmes données sur différents claviers. On pourrait croire qu'il s'agirait simplement d'échanger ces mêmes données par transmissions électroniques ou à l'aide de l'utilisation de disques compacts et de disquettes. Toutefois, les données emmagasinées peuvent décrire les mêmes cibles mais dans un jargon incompatible. Cette situation démontre que les forces militaires modernes doivent prendre des mesures additionnelles pour espérer réaliser l'efficacité opérationnelle réclamée par les adeptes des systèmes d'information.

Le but de ce projet est d'évaluer la progression et le potentiel prometteur de la technologie XML dans l'atteinte rapide de l'interopérabilité des systèmes d'information pour les Forces terrestres.

Cette recherche démontre le bien fondé des ressources investies par les Forces terrestres de concert avec les forces alliées pour promouvoir l'élaboration d'un vocabulaire de langage de balisage militaire MilML qui saura tirer profit des incessants développements de la technologie XML. L'expérimentation étudiera la possibilité de rallier le nouveau système de commandement et contrôle reposant sur le Operational Planning Environment and Reference Application (OPERA) avec l'ancienne technologie du Communication Management System (CMS) du système IRIS. Cependant d'autres barrières doivent être franchies avant d'appliquer directement la technologie XML dans une situation semblable.

## **Sommaire Exécutif**

### **XML – L’avenir de l’interopérabilité**

#### **Une solution pour les systèmes d’information des Forces terrestres**

« La numérisation du champ de bataille est inévitable ». Les forces militaires possèdent de complexes systèmes d’information très spécialisés mais conçu pour un rôle spécifique. Malgré leur sophistication, comment se fait-il que les machines doivent encore compter sur le soutien humain pour demeurer informées de la situation avoisinante? C’est simple, l’interopérabilité entre les systèmes d’information est au stade théorique. On se doit de renverser la situation et de promouvoir tout effort d’interopérabilité entre les machines et enfin libérer les soldats du fardeau de la cueillette d’information manuelle.

Les technologies réseau font rapidement avancer le dossier d’interopérabilité et de globalisation. De concert avec l’OTAN, les Forces terrestres canadiennes s’engagent dans cette direction avec comme objectif l’établissement de ponts ralliant ces îlots informatiques. L’avenir est prometteur pour les nouveaux systèmes OPERA, Athene et SAS qui sont sur le point d’être livrés. Ils se sont servis du modèle de données LFDM favorisant l’interopérabilité au niveau des FT. La situation est moins certaine pour les systèmes construits avec d’anciennes technologies sans le modèle de référence des FT tel que le CMS.

Ce document survole les technologies réseau CORBA, COM et DCOM, RMI pour fin d’interopérabilité mais met l’accent sur les technologies du XML. Il mettra en lumière la robustesse et la souplesse d’utilisation du XML qui en font un outil remarquable pour l’environnement militaire. Le milieu de l’informatique crée sans cesse de nouvelles applications pour XML qui peuvent être adoptées par le milieu militaire. En adoptant cette technologie, les gérants de systèmes d’information (SI) militaires pourront compter sur une expertise professionnelle assurée. Le XML pourra relever la perception

d'insécurité en rendant les services de sécurisation essentiels dans la gestion de réseaux et de protection des droits d'accès à l'information.

En créant des vocabulaires plus personnalisés tel que le ebXML, le ChemXML, et le RosettaNet, le monde dynamique des affaires a traduit et codifié leur processus de production de façon à faciliter leurs échanges commerciaux et augmenter leur rentabilité. De même, les forces militaires augmenteront leur efficacité opérationnelle en continuant leurs efforts de développement pour le vocabulaire militaire MilML qui vise à structurer les échanges de données correspondant aux opérations militaires. C'est dans cette perspective que l'on pourra rapidement atteindre l'interopérabilité des FT.

Les difficultés liées à l'interopérabilité ne repose malheureusement pas seulement sur le choix d'une technologie. Les efforts de création d'une interface entre OPERA et CMS ont permis d'identifier d'autres défis tout aussi important à relever :

- L'atteinte d'un niveau de connaissance suffisante sur le fonctionnement des SI demande le soutien actif de leur bureau d'appartenance;
- Les systèmes d'exploitation doivent être capables d'utiliser des outils de développement modernes;
- Une description de la structure et du fonctionnement des bases de données;
- Un modèle de référence de données qui peut être partager par les deux SI;
- La capacité d'extraire et d'injecter des données dans les bases de données.

L'apport du XML est certain pour les deux derniers points. Pour les autres, le succès repose sur l'engagement du personnel qui est imbriqué dans le soutien de son SI. L'interopérabilité doit être un facteur à considérer lors de la livraison de leur SI. Malgré ces défis l'interopérabilité est à portée de main car on peut compter sur la validité de la technologie actuelle.

## Remerciements

Nous tenons spécialement à remercier le département d'informatique de la faculté des sciences et génie de l'Université Laval qui a mis sur pied un cours compressé d'introduction au langage de balisage XML, Pierre D'Anjou et Marie Théberge du groupe CGI de Québec pour la description détaillée du système OPERA, souligner le soutien continu de Johanne Blais, du major Jean-René Couture et du lieutenant colonel Jacques Hamel tous les trois de la direction des communications et information terrestre, et finalement l'équipe du projet IRIS en particulier l'adjudant Francis pour avoir fourni le matériel d'expérimentation.

## TABLE DES MATIÈRES

Avertissement	i
Abstrait	ii
Sommaire	iii
Remerciements	v
Table des matières	vi
Liste des figures et des tables	x
Chapitre 1 - INTRODUCTION	1-1
1.0 Généralités	1-1
1.1 But	1-2
1.2 Portée	1-2
1.3 Approche	1-2
1.4 L'interopérabilité militaire internationale	1-3
1.5 L'interopérabilité des FT	1-3
Chapitre 2 – L'interopérabilité et les technologies actuelles	2-1
2.0 Généralités	2-1
2.1 Les différentes couches d'interopérabilité	2-2
2.1.1 Interopérabilité mémoire	2-2
2.1.2 Interopérabilité du code source	2-3
2.1.3 Interopérabilité de Type d'information	2-4
2.1.4 Interopérabilité de protocole de transmission	2-5
2.2 Efficacité en réseaux des technologies d'interopérabilité	2-6
2.3 Les technologies existantes en interopérabilité	2-7
2.4 COM et DCOM	2-8
2.5 CORBA	2-9
2.6 Java et RMI	2-10
Chapitre 3 – Le XML et ses composantes	3-1
3.0 Le XML - Un historique	3-1
3.0.1 SGML	3-1
3.0.2 HTML	3-1
3.0.3 Le W3C	3-2
3.1 Introduction au XML	3-2
3.1.1 La syntaxe du XML	3-5
3.2 Les composantes du XML	3-7
3.2.1 DTD	3-7
3.2.1.1 Définition de DTD	3-7

3.2.2	Schemas XML	3-9
3.2.2.1	Les éléments et attributs d'un schéma XML	3-10
3.2.2.2	Les Types de données	3-11
3.2.2.2.1	Les types simples	3-11
3.2.2.2.2	Les types complexes	3-12
3.2.2.2.3	Les types listes	3-12
3.2.2.3	Les espaces de nom ou "namespaces"	3-13
3.2.3	Document Object Model (DOM) et Simple API for XML (SAX)	3-14
3.2.3.1	Comparaison entre DOM et SAX	3-14
3.3	L'interopérabilité et le XML	3-15
3.3.1	La plateforme XML	3-15
3.3.2	Simple Object Access Protocol (SOAP)	3-16
Chapitre 4 – Sécurité entourant le XML		4-1
4.0	Généralités	4-1
4.1	Protection d'instances d'utilisation du XML	4-1
4.2	XML Key Management Specification (XKMS)	4-2
4.2.1	XML Signature	4-3
4.2.2	XML Encryption	4-3
4.3	XML pour accroître la sécurité	4-4
4.3.1.	Diverse Applications XML sécuritaires	4-6
4.4	Sommaire sur la sécurité du XML	4-6
Chapitre 5 – Les vocabulaires du XML		5-1
5.0	Généralités	5-1
5.1	Identification des raisons motivant le développement des vocabulaires par couche de l'industrie.	5-1
5.2	Les vocabulaires répertoriés	5-2
5.2.1.	ChemXML	5-3
5.2.2.	MathML	5-3
5.2.3.	TranXML	5-4
5.2.4.	HR-XML	5-4
5.2.5.	ebXML	5-5
5.2.6.	RosettaNet	5-5
5.2.7.	XBRL	5-6
5.2.8.	LegalXML	5-6
5.3	Sommaire	5-6
Chapitre 6 – MilML – Un vocabulaire structuré pour l'environnement militaire		6-1
6.0	Généralités	6-1
6.1	Modèle d'échange de données commun – LC2IEDM/LFDM	6-1

6.2. MilML	6-2
6.2.1. Le MilML construit pour les opérations tactiques	6-2
6.2.2. Les spécifications techniques du MilML	6-3
6.4.3. La structure physique du MilML- entités	6-3
6.4.4. Les OBJETS TYPE et les OBJETS ITEM du MilML	6-3
6.5 Sommaire sur le MilML	6-5
Chapitre 7 – Une solution pour les systèmes d’information des forces terrestres	7-1
7.0 Généralités	7-1
7.0.1 Operational Data Store (ODS)	7-2
7.1 Description du problème d’interopérabilité entre Opéra et CMS	7-4
7.2 Description du système d’information OPÉRA	7-6
7.2.1 Les composantes d’OPERA	7-7
7.2.2 L’interface utilisateur	7-8
7.2.3 La structure de la base de données	7-9
7.2.4 Le flux de l’information	7-10
7.2.5 Extracteur de données (Data pump)	7-11
7.2.6 Le gestionnaire de mise à jour	7-11
7.2.7 La couche d’information et de transformation (ITL)	7-12
7.2.8 Le XML Transformer	7-13
7.2.9 La synchronisation	7-14
7.3 Description du système d’information CMS	7-15
7.3.1 Généralités	7-15
7.3.2 Le système IRIS	7-15
7.3.3 Le système CMS	7-16
7.3.4 L’information capturée manuellement par CMS	7-17
7.3.5 Technologie utilisée dans l’environnement CMS	7-18
7.3.6 Effort requis pour l’interopérabilité	7-19
7.4 Modèle théorique d’interopérabilité OPERA/CMS	7-20
7.4.1 Généralités	7-20
7.4.2 Présentation des options	7-21
7.4.2.1 Utilisation de la DATA PUMP d’OPERA	7-21
7.4.2.2 Conception d’un injecteur de données	7-22
7.4.3 Choix d’une option	7-23
7.4.4 Sélection d’un langage	7-23
7.4.5 Design du programme	7-24
7.4.5.1 Commentaires de spécification de l’injecteur de données	7-24
7.4.5.2 Pseudo-Code de l’application	7-25
7.4.6 L’analyse des tables des systèmes OPERA et CMS	7-26
7.4.6.1 Analyse des tables de CMS	7-26
7.4.6.2 Analyse des tables d’OPERA	7-27
7.4.6.3 Le mappage et la correspondance des tables	7-29
7.5 Expérimentation avec le modèle théorique	7-30
7.5.1 Généralités	7-30
7.5.2 Description du banc d’essai	7-30



*Table des matières*

7.5.3	Expérimentation avec OPERA	7-31
7.5.4	Expérimentation avec CMS	7-31
7.6	Recommandations liées à l'interopérabilité entre Opéra et CMS	7-34
Chapitre 8 - Conclusion et Recommandations		8-1
8.0	Conclusion	8-1
8.1	Généralités	8-1
8.2	Solutions technologiques aux problèmes d'interopérabilité	8-1
8.3	Le potentiel du XML	8-2
8.4	L'aspect sécurité	8-3
8.5	L'expérimentation	8-3
8.6	Recommandations	8-4
8.6.1	Utilisation du XML en interopérabilité	8-4
8.6.2	Modèles de données	8-5
8.6.3	Modernisation des systèmes anciens	8-5
8.6.4	Contrôle de l'information	8-5
Annexe A – Convention Logique-à-physique des Noms		A-1
Annexe B –Entités MilML		B-1
Annexe C – Interface CMS et CMMS		C-1
Annexe D - Interrelation Plan et BFSM		D-1
Annexe E – Interrelation et rétroaction entre PLAN et BFSM		E-1
Annexe F – Tables de base de données physique – OPERA		F-1
Appendice 1 – Tables de base de données Logique – OPERA		F1-1
Annexe G – Tables de base de données physique – CMS		G-1
Annexe H – Rapport – XML Application for CMS/OPERA Compatibility		H-1
Glossaire		G-1
Bibliographie		b-1

---

Liste des figures et des tables

Figure 1-1	Schéma de l'architecture globale AIME	1-4
Figure 2-1	Modèle de couches d'interopérabilité	2-2
Figure 2-2	Les tables de fonctions virtuelles de COM	2-3
Figure 2-3	Utilisation de la librairie COM	2-3
Figure 2-4	L'utilisation de l'interface IDL de CORBA	2-4
Figure 2-5	Le modèle trois couches du protocole RMI de Java	2-5
Tableau 2-1	L'intégration des technologies d'interopérabilité	2-7
Figure 2-6	Le protocole de DCOM	2-8
Figure 2-7	Invocation CORBA	2-9
Figure 2-8	Le Remote Method Invocation	2-10
Figure 3-1	Affichage d'un document HTML	3-3
Figure 3-2	Affichage de base d'un document XML	3-4
Figure 3-3	Types de données prédéfinis de XML Schema	3-11
Figure 3-4	Structure d'un message SOAP	3-16
Figure 4-1	Vulnérabilité des documents XML échangés	4-1
Figure 4-2	Document XML protégé sauf contre la vulnérabilité aux bris de transmission	4-4
Figure 4-3	Les interactions SAML	4-5
Figure 6-1	Modèle de données des forces terrestres	6-2
Figure 6-2	Hiéarchie des objets typiques du « LC2IEDM »	6-4
Figure 6-3	Exemple de l'interrelation de la structure MilML	6-5
Figure 7-1	Modèle Multi-Systèmes Interconnectés	7-1
Figure 7-2	Modèle Multi-systèmes concentré sur une base de données	7-2
Figure 7-3	ODS- La fondation des échanges de données pour les FT	7-2
Figure 7-4	Problématique d'interopérabilité entre OPERA et CMS	7-4
Figure 7-5	L'environnement électronique intégré	7-6
Figure 7-6	L'interface d'OPERA	7-8
Figure 7-7	Le modèle relationnel orienté-objet de LFDM et OPERA	7-9
Figure 7-8	Le flux d'information au travers des couches de transfert de données	7-10
Figure 7-9	Fonctionnement de l'extracteur de données	7-11
Figure 7-10	La synchronisation	7-14
Figure 7-11	Composantes du Système IRIS	7-15
Figure 7-12	Architecture du Système IRIS	7-16
Figure 7-13	Terminal de données Portatif(TDP)	7-17
Figure 7-14	Concept d'utilisation de la DATA PUMP Java d'OPERA avec CMS	7-21
Figure 7-15	L'injecteur de données XML	7-22
Figure 7-16	Représentation relationnelle des tables de CMS	7-26
Tableau 2-1	Table VEHICLE de CMS	7-27
Tableau 2-2	Table UNIT d'OPERA et tableau 7-3 Table UNIT de CMS	7-28
Figure 7-17	Effectuer la correspondance entre les tables des deux systèmes	7-29
Figure 7-18	Processus de chargement d'application propre à SCO UNIX	7-32

DISCLAIMER AVERTISSEMENT

<p>This report was written by a student on the Land Force Technical Staff Programme at the Royal Military College of Canada, Kingston, Ontario. This report is an unofficial document and may contain errors. The views expressed in the report are those of the authors and do not necessarily reflect the opinion or policy of the Royal Military College, the Canadian Forces, the Department of National Defence or the Government of Canada</p> <p>© Her Majesty the Queen as represented by the Minister of National Defence.</p>	<p>Le présent rapport a été rédigé par un étudiant du Programme d'état-major technique de la Force terrestre du Collège militaire royal du Canada, Kingston, Ontario. Ce rapport est un document non officiel et pourrait renfermer des erreurs. Les opinions exprimées dans le présent rapport sont celles de l'auteur et ne représentent pas nécessairement l'opinion ou la politique du Collège militaire royal, des Forces armées canadiennes, du ministère de la Défense nationale ou du gouvernement du Canada.</p> <p>© Sa Majesté la Reine représentée par le ministre de la Défense nationale.</p>
---	---

## Chapitre 1 – Introduction

### 1.0 Généralités

Le débat d'évolution ou de révolution dans les affaires militaires a pris place depuis quelques années déjà. Même si cette question n'est pas encore réglée, il n'en demeure pas moins que l'avènement de nouvelles technologies au niveau de l'électronique, de l'informatique et des télécommunications créent des conditions favorables pour la numérisation du champ de bataille, dont un aspect important est l'échange d'information. Ceci se révèle maintenant être problématique en raison de l'introduction chaotique de la numérisation dans les systèmes d'armes et d'information des Forces terrestres (FT).

Les Directives stratégiques sur les opérations et les ressources 2002 (DSOR 2002), visent à établir une capacité de soutien au commandement en numérisant l'environnement au niveau de brigade, contribuant à atteindre l'interopérabilité avec nos principaux alliés<sup>1</sup>.

On a assisté à une numérisation graduelle mais sans encadrement des systèmes d'armes au sein des FT. Cette absence de stratégie a eu pour conséquence de produire des systèmes d'information (SI) orphelins qui sont incapables d'échanger l'information vitale rendant plus décisifs les succès sur le champ de bataille

Malgré toutes ces nouvelles technologies, on a recours à des communications vocales pour l'échange d'information tactique autrement transmissible électroniquement. La capture manuelle de saisie de données de manière répétitive et redondante cause des délais importants dans la prise de décision tout en consommant des ressources en personnel déjà insuffisantes.

Heureusement, la tendance actuelle favorise l'introduction de systèmes d'information dont l'architecture est basée sur l'interopérabilité. Dans cette lancée, on réalise que des efforts doivent être faits afin de rendre interopérables et compatibles les SI "legacy" toujours en place.

L'industrie a reconnu le problème d'interopérabilité et des solutions existent. La complexité et la difficulté d'implantation varie d'une technologie à l'autre, mais des progrès en ce sens ont été réalisés ces dernières années et l'une d'elles est simple, efficace et suscite un vif intérêt au niveau de la communauté informatique et du commerce électronique. Il s'agit du Langage de Balisage Extensible ou "eXtensible Markup Language" (XML).

---

<sup>1</sup> Chef d'état-major de l'Armée de Terre, Directives stratégiques sur les opérations et les ressources 2002, Chapitre 3, Section 2, Annexe A, page 4.

## 1.1 But

Le but de ce projet est d'évaluer la progression et le potentiel prometteur de la technologie XML dans l'atteinte rapide de l'interopérabilité des systèmes d'information pour les FT.

## 1.2 Portée

Ce projet parrainé par la direction des communications et de l'information – terrestre (DCIT) permettra d'établir les mérites et l'obligation pressante de faire de l'interopérabilité la priorité dans le développement continu de tous les SI envisagés pour les FT. Parmi les interfaces potables, les FC de concert avec ses alliés ont retenu de préférence le XML comme outil de base pour réaliser l'interopérabilité. Les efforts en ce sens ont déjà défini les structures communes reposant sur les particularités du XML. DCIT a pris en main la tâche de s'assurer que toutes les applications sur le point d'être livrées par les FT, entre autre l'Athene Tactical System (ATS) et Situational Awareness System (SAS), soient compatibles avec la technologie XML. La balance du fardeau résidera dans la conversion ou la façon de rallier les autres applications technologiquement plus âgés vers une structure d'échange d'information commune.

Notre intérêt sera d'expérimenter ce concept en explorant les défis liés à l'échange d'information entre deux applications des FT soit OPERA (Operational Planning Environment and Reference Application) et le système de gestion des communications (SGC). Dans ce rapport, l'expression anglaise Communication System Management (CMS) sera utilisé au lieu de SGC.

## 1.3 Approche

L'objectif de ce travail sera accompli en survolant les technologies courantes utilisées dans les travaux d'interopérabilités, puis d'identifier les mérites liés au XML qui est l'outil adopté par l'OTAN et les FT. L'étude évaluera les termes par lesquels la technologie XML pourra être utilisée pour permettre les transactions entre deux SI construit avec des technologies différentes.

## 1.4 L'interopérabilité militaire internationale

En ce qui concerne l'interopérabilité, le Canada a joint les efforts de l'OTAN après l'Exercice Harmony qui a eu lieu en 1993. Les efforts d'interopérabilité avaient débutés auparavant avec le Quadrilateral Interoperability Programme (QIP) en 1984<sup>2</sup> visant le ralliement des niveaux de brigades à corps. L'objectif final du QIP a été atteint en décembre 1998. Parallèlement, un autre programme complémentaire avait débuté: le Battlefield Interoperability Program (BIP) en 1995<sup>3</sup> et qui visait le ralliement du niveau de bataillon en descendant. Ce programme s'est lui aussi terminé avec succès après le test de l'interface développé pour l'Exercice CONCORDIA en novembre 1997.

Les succès de ces programmes ont justifié leur remplacement en 1998 par le Multilateral Interoperability Programme (MIP) qui implique maintenant le Canada, la France, l'Allemagne, l'Italie, l'Angleterre, l'Australie et les États-Unis. Le but du MIP est de réaliser l'interopérabilité internationale grâce à un système automatisé d'échange d'information entre les systèmes de commandement et contrôle des forces terrestres des nations participantes<sup>4</sup>. Cet objectif englobe les SI en partant du corps jusqu'au plus bas niveau, incluant les opérations combinées et interarmées grâce à l'avancement de la numérisation dans le milieu international.

Le MIP servira à capturer les ententes sur les processus utilisés pour endosser toute proposition apportée<sup>5</sup> pour les IER, pour l'Allied Data Publication No 3 (ADatP-3) ainsi que le Land Command and Control Information Exchange Data Model (LC2IEDM). Le LC2IEDM est la nouvelle terminologie qui remplace le terme Army Tactical Command and Control Information System ATCCIS. Au travers du model LC2IEDM de l'OTAN, on cherche à produire une vue corporative représentant l'échange d'information multinational<sup>6</sup>.

## 1.5 L'interopérabilité des FT

Ainsi on décrira l'échange d'information dans l'environnement de commandement et contrôle. Pour que cette information soit aisément automatisée, on doit se conformer spécifiquement à la nomenclature et à la structure de l'information partagée<sup>7</sup>. L'adaptation du LC2IEDM de l'OTAN au contexte canadien, a donné lieu au Land Force Data Model (LFDM) sous la tutelle du Canadian Army Integrated Management Environment (AIME) project. L'architecture fonctionnelle interne du AIME se base largement sur le « Distributed Component Object Model » (DCOM), Java et XML pour permettre l'échange d'information entre plusieurs processus distribués entre

---

<sup>2</sup> Multilateral Interoperability Programme (MIP), MIP Interface Operating Procedures - FR-PWG – Version A3, 21 septembre 2001, page 10.

<sup>3</sup> Idem.

<sup>4</sup> Idem.

<sup>5</sup> Idem, page 12.

<sup>6</sup> DLIR, Army Integrated Management Environment (AIME) Overall Technical Architecture v 1.7, 12 March 2001, page 4.

<sup>7</sup> Idem, page 5.

des serveurs éloignés<sup>8</sup>. Une représentation graphique de l'environnement des forces terrestres est présentée à la figure 1-1.

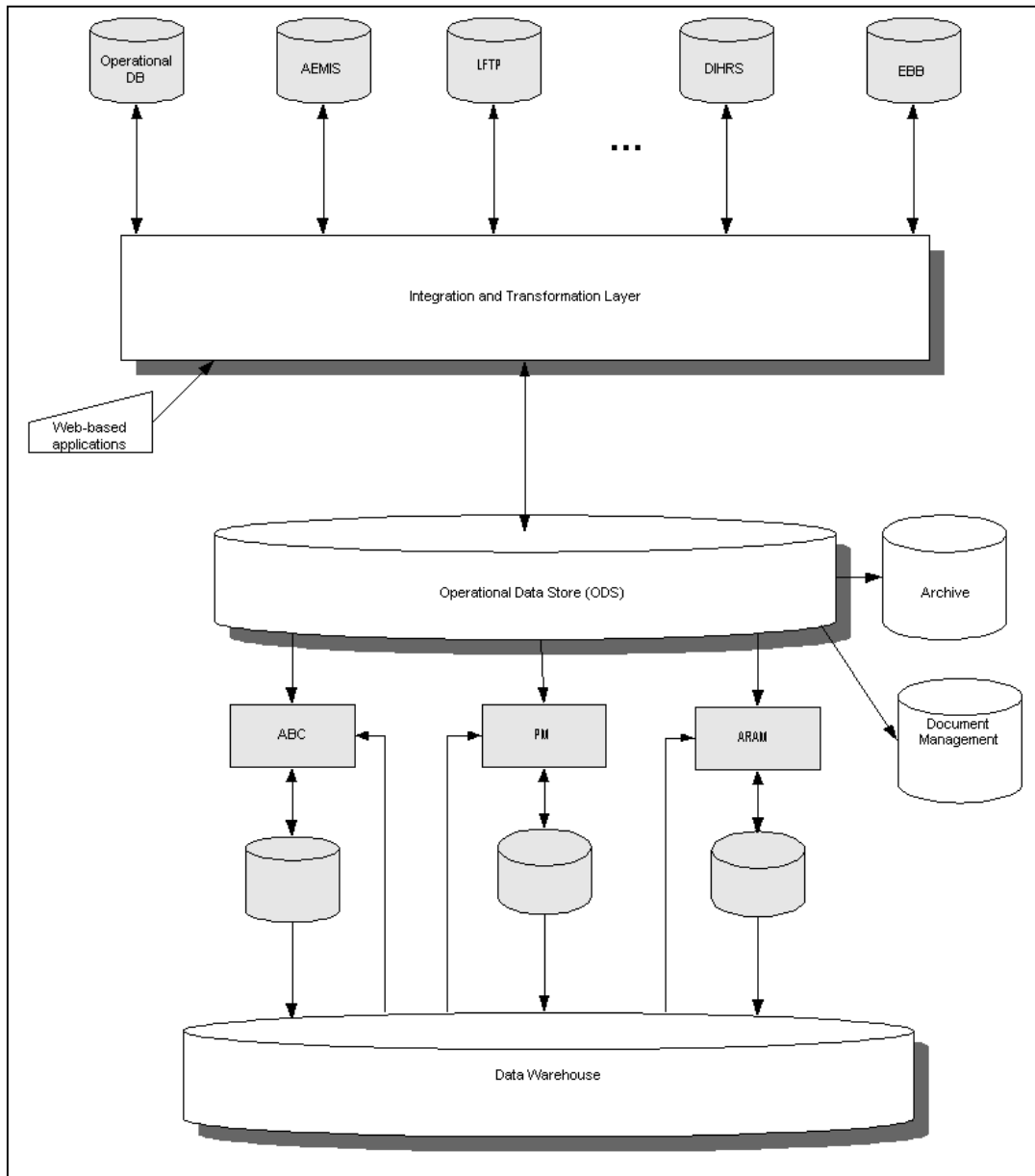


Figure 1-1 Schéma de l'architecture globale AIME<sup>9</sup>

<sup>8</sup> Idem, page 1.  
<sup>9</sup> Idem, page 24.

Pour l'expérimentation, on partira de l'application OPERA pour démontrer la valeur du XML pour l'interopérabilité. Le logiciel OPERA est la nouvelle version du coffret électronique exprimé dans la figure 1-1 en anglais « EBB » pour Electronic Battle Box. Ce logiciel fera parti d'une suite développée pour le commandement et contrôle qui inclut « CFCS », « LFCS », « PDALF » et « TMHS ». Il contient des données sur les procédures d'état-major. Il assistera à la création des organigrammes et à la planification des log/lift planner. OPERA sera discuté plus longuement au chapitre 7.

L'autre application utilisée pour cette expérimentation est le CMS, un logiciel faisant partie de la suite produite pour le STCCC. Cette application gère le système IRIS de façon à maintenir l'efficacité des ressources de communication déployées pour soutenir les commandants et leurs systèmes de commandement et contrôle<sup>10</sup>. Ce logiciel est basé sur une ancienne technologie.

Ce travail de recherche repose en grande partie sur l'étude des tendances des nouvelles technologies d'interopérabilité et en particulier du XML et ses bénéfices pour les FT.

---

<sup>10</sup> BP STCCC/SLI, Aperçu du Système IRIS, C-53-750-000/MA-001, 26 mars 1999, page 4-2-1.



## Chapitre 2 – L'interopérabilité et les technologies actuelles

### 2.0 Généralités

L'interopérabilité des SI signifie la capacité de composantes informatiques de s'intégrer entre elles, même si elles ont été conçues et développées de manière différente et dans des langages différents. Au sein du Ministère de la défense nationale, l'interopérabilité des SI est définie comme « la capacité de partager de l'information. L'environnement d'information intégré à pour but de partager l'information au travers de tous les éléments, fonctions et niveaux de commandement »<sup>1</sup>.

Le ministère a ainsi établi quatre niveaux d'interopérabilité<sup>2</sup>:

- a. Niveau 1 – Échange de message de base (ex: Officier de liaison)
- b. Niveau 2 – Échange de messages structurés/formatés (USMTF, ADAP-P3)
- c. Niveau 3 – Structures de données/information commune (ex bases de données partagées par plusieurs applications)
- d. Niveau 4 – Intégration complète (ex Applications logicielle distribuées)

L'interopérabilité entre SI est possible grâce à des technologies informatiques qui établissent un lien entre ces composantes appartenant à des applications différentes. À première vue, toutes les technologies existantes fonctionnent de façon plus ou moins semblable. Dans un contexte de boîte noire, il y a en effet peu de différences, car le résultat est semblable. Cependant chaque technologie a ses propres forces et faiblesses. Les technologies existantes et ayant le plus d'intérêt dans le cadre de notre recherche sont:

- a. Langage de balisage étendu (XML) (sera discuté au chapitre 3);
- b. Component Object Model (COM) et Distributed Component Object Model (DCOM);
- c. Common Object Request Broker Architecture (CORBA); et
- d. Remote Method Invocation (RMI) de Java.

---

<sup>1</sup> Groupe CGI, "Introduction to ATCCIS Information Session presented to Land Forces Staff", Mars 2000

<sup>2</sup> Idem

## 2.1 Les différentes couches d'interopérabilité

Avant d'examiner les technologies reliées à l'interopérabilité, il faut comprendre comment l'interopérabilité prend place et entre quel type de composante elle agit. Nous présentons un modèle multi-couche qui représente quatre degrés d'interopérabilité au niveau technique. Nous verrons ces couches au travers d'exemples touchant les technologies COM/DCOM, CORBA et Java.

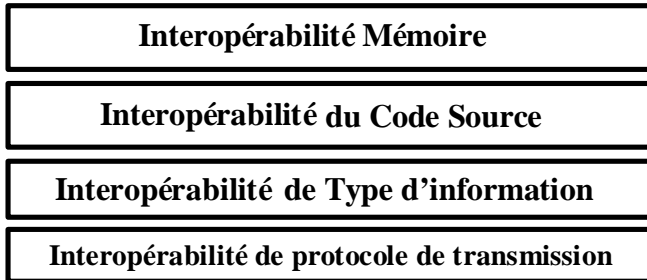


Figure 2-1 Modèle de couches d'interopérabilité(traduit et redessiné)<sup>3</sup>

### 2.1.1. Interopérabilité mémoire

En utilisant une représentation normalisée en mémoire, toutes les composantes doivent la respecter afin d'atteindre l'interopérabilité, cette technologie offre une excellente performance en vitesse d'exécution.

« Les composantes informatiques coordonnent entres-elles l'allocation des objets en mémoire afin de coexister sans collision. »<sup>4</sup> Cette technologie requiert de plus grands efforts et est souvent limitée au système d'exploitation.

La technologie COM par exemple normalise la représentation en mémoire des références à des objets basés sur des tables de fonctions virtuelles de C++.

---

<sup>3</sup> Box, Don, "Lessons from the Component Wars: An XML Manifesto",  
<<http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnxml/html/xmlmanifesto.asp>>,  
Septembre 1999 Copyright © 2001 Microsoft Corporation, One Microsoft Way, Redmond, Washington  
98052-6399 U.S.A. All rights reserved.

<sup>4</sup> Guerrero Ignacio, "OPENGIS XML- BASED WEB MAP INTERFACES",  
<[http://www.canri.nsw.gov.au/activities/dsd2001/workshop2/wkshp2\\_intergraph.htm](http://www.canri.nsw.gov.au/activities/dsd2001/workshop2/wkshp2_intergraph.htm)>, 13 mai 2002.

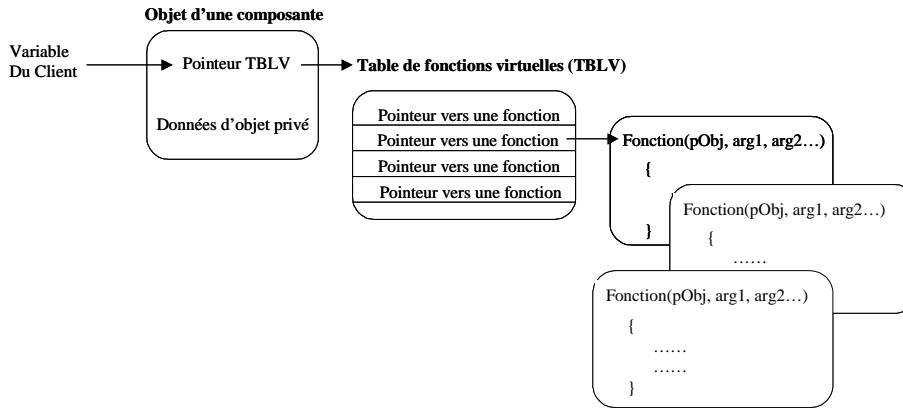


Figure 2-2 Les tables de fonctions virtuelles de COM (traduit et redessiné) <sup>5</sup>

### 2.1.2. Interopérabilité du code source

Les développeurs programment en utilisant une interface de programmation d'application ou *application programming interface* (API). Un code source développé sur une API en particulier pourra être compilé sur une API d'un autre manufacturier.

Par exemple, « COM rend disponible aux API ses services via la librairie COM. Cette façon de procéder rend possible la compilation du même code source sur des plateformes différentes, comme Windows NT, Windows 95, Linux, ou Solaris. »<sup>6</sup>

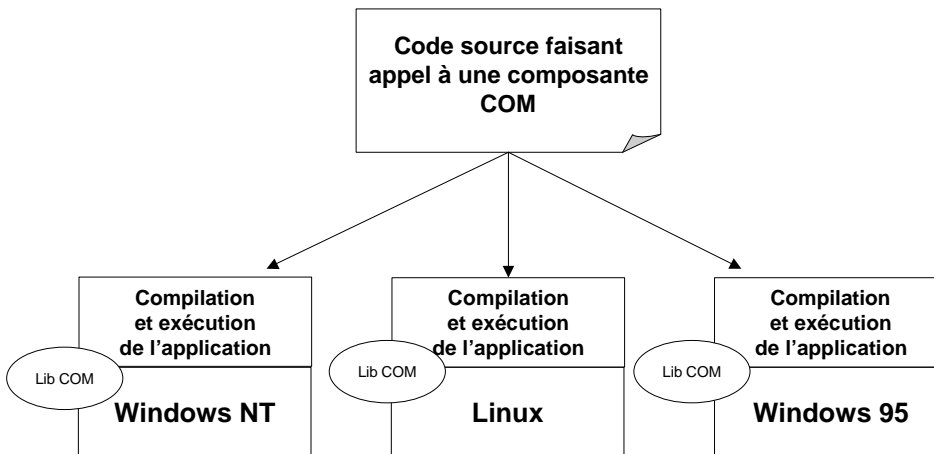


Figure 2-3 Utilisation de la librairie COM

<sup>5</sup> Kindel, Charlie and Williams, Sara, "The Component Object Model: A Technical Overview", <[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn\\_compr.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn_compr.asp)>, Octobre 1994.

<sup>6</sup> Box, Don, "Lessons from the Component Wars: An XML Manifesto", <<http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnxml/html/xmlmanifesto.asp>>, Septembre 1999 Copyright © 2001 Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399 U.S.A. All rights reserved.

### 2.1.3 Interopérabilité de Type d'information

Les technologies utilisées en interopérabilité doivent fournir une façon normalisée de représenter le type d'information qu'elles transportent aux programmeurs et aux infrastructures informatiques.

Par exemple, « CORBA fourni une interface de définition de langage ou *interface definition language* (IDL) qui permet de décrire les objets d'une manière neutre. En définissant dans cette IDL tous les types de données accessibles, un objet CORBA peut être accessible à partir de n'importe quel langage de programmation qui bénéficie du support "Object Request Broker" (ORB). La technologie COM a elle aussi une IDL qui est plus ou moins équivalente à celle de CORBA. »<sup>7</sup>

Certaines technologies n'utilisent pas cette couche d'interopérabilité. « Avec Java, toutes les composantes Java respectent un format de classes auto-descriptif, aucun support de type d'information n'est nécessaire. »<sup>8</sup>

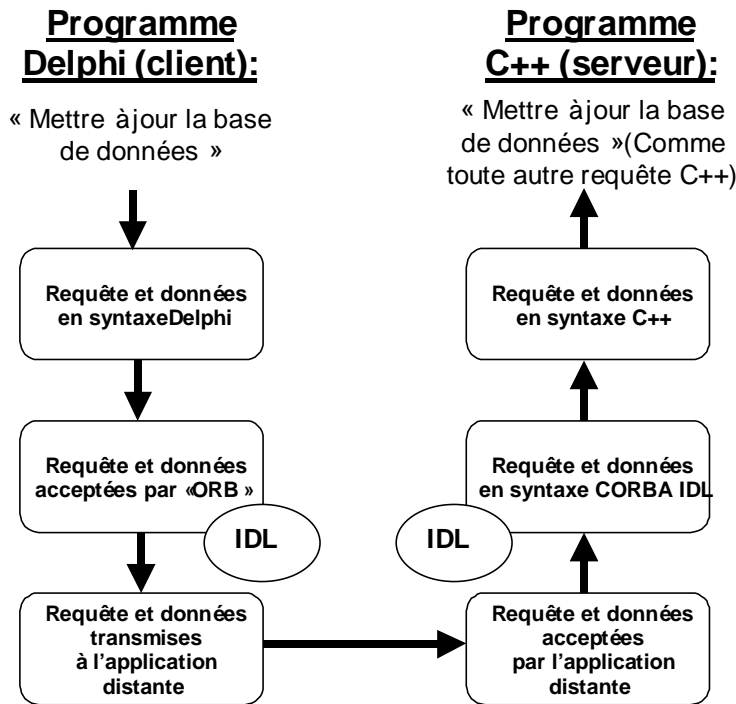


Figure 2-4 L'utilisation de l'interface IDL de CORBA (traduit et redessiné)<sup>10</sup>

<sup>7</sup> Box, Don, "Lessons from the Component Wars: An XML Manifesto", <<http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnxml/html/xmlmanifesto.asp>>, Septembre 1999, Copyright © 2001 Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399 U.S.A. All rights reserved.

<sup>8</sup> Idem

<sup>9</sup> DISA/JIEO, "DII COE Distributed Applications Series - Recommendations for Using DCE, DCOM, and CORBA Middleware", 13 avril 1998, Figure 1, page 14

<sup>10</sup> Idem

### 2.1.4 Interopérabilité de protocole de transmission

Les technologies réseaux sont de plus en plus présentes au sein des applications informatiques. Même les applications logicielles distribuées doivent bénéficier de ces technologies.

Les technologies COM et DCOM utilisent un mécanisme appelé *Distributed Computing Environment Remote Procedure Call* (DCE RPC) qui encode et transporte les données entre les machines hôtes.

CORBA utilise le protocole *Internet Inter-ORB Protocol* (IIOP). Ce protocole se sert du protocole TCP pour le transport des données.

Java supporte IIOP et CDR en plus de son propre protocole *Remote Method Invocation* (RMI) qui permet à n'importe quelle application Java d'appeler une méthode distante et résidente sur une autre machine comme si elle était locale à elle-même. Java utilise ainsi le protocole *Java Remote Method Protocol* (JRMP) qui permet au RMI d'utiliser les protocoles standards TCP ou HTTP.

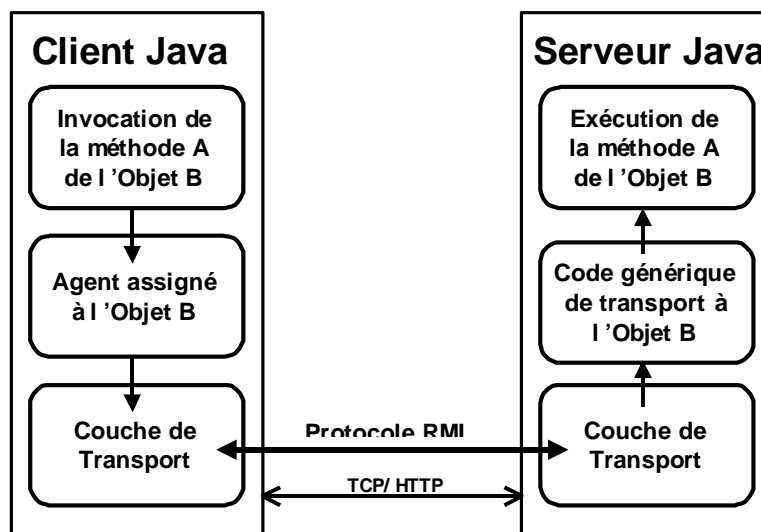


Figure 2-5 Le modèle trois couches du protocole RMI de Java (traduit, adapté et redessiné)

<sup>11</sup> Sun Microsystems, "Java™ Remote Method Invocation – Pure Java Distributed Computing", <[http://java.sun.com/marketing/collateral/rmi\\_ds.html](http://java.sun.com/marketing/collateral/rmi_ds.html)>, 18 mai 2002, Copyright 1994-2002 Sun Microsystems, Inc.

## 2.2 Efficacité en réseaux des technologies d'interopérabilité

Pour exploiter la puissance de ces technologies, il faut envisager leur utilisation dans un contexte réseau. DCOM, CORBA et Java ont tous des protocoles de communication qui leur sont propres. Le TCP et HTTP sont d'autres protocoles de communication qui sont utilisés. Ils sont surtout efficaces en raison de leur facilité à passer au travers des "Firewalls" et ainsi transmettre facilement leurs données.

Dans ce contexte, le TCP/HTTP est souvent utilisé comme moyen d'atteindre l'ordinateur avec lequel on veut communiquer et de passer sans être bloqué. Cependant, passer par le HTTP nécessite un temps d'exécution non négligeable afin de remplir les fonctions d'interopérabilité. Par exemple, « le protocole RMI de Java est au moins d'un ordre de magnitude plus lent lorsqu'il utilise le protocole HTTP que lorsqu'il passe directement par un "socket" de communication »<sup>12</sup>.

---

<sup>12</sup> Sun Microsystems, "RMI System Overview - Chapter 3.5: RMI Through Firewalls Via Proxies", section 3.5.5, <<http://java.sun.com/j2se/1.4/docs/guide/rmi/spec/rmi-arch6.html>>, 18 mai 2002, , Copyright 1994-2002 Sun Microsystems, Inc.

### 2.3 Les Technologies existantes en interopérabilité

Comme mentionné auparavant, cette section examinera plus en détail les trois technologies les plus utilisées en interopérabilité, laissant la discussion à propos du XML au chapitre trois.

Voici tout d'abord un tableau comparatif d'identification des technologies au niveau des couches d'interopérabilité, incluant le XML, qui résumera bien la section précédente:

Couche d'interopérabilité	COM	Java	CORBA	XML
En mémoire	L'API de COM	Le langage de programmation Java	Object Request Broker (ORB) et Portable Object Adapter (POA)	W3C Document Object Model (DOM) API simple du XML (SAX)
Type d'information (fichiers texte)	L'IDL de COM	Le langage de programmation Java	L'IDL du Object Managment Group (OMG)	DTD, Schemas XML
Type d'information (fichiers binaires)	Librairies de type de données	Les fichiers .class	aucune	DTD, Schemas XML
Type d'information (niveau d'API)	Les librairies LoadTypeLib, ItypeLib, et autres	java.lang.reflect	Interface résidente	Schemas XML, tout " <i>parser</i> " XML fonctionne
protocole de communication	DCOM au travers des protocoles TCP, SPX, etc...	RMI/JRMP ou RMI/IIOP ou RMI/HTTP	IIOP au travers du protocole TCP	XML au travers des protocoles HTTP ou TCP, ou un protocole de messagerie

Tableau 2-1 L'intégration des technologies d'interopérabilité (adapté et redessiné) <sup>13</sup>

<sup>13</sup> Box, Don, "Lessons from the Component Wars: An XML Manifesto", <http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnxml/html/xmlmanifesto.asp>, Septembre 1999, , Copyright © 2001 Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399 U.S.A. All rights reserved.

## 2.4 COM et DCOM

Cette technologie d'interopérabilité est un produit de la corporation Microsoft, et Orienté-Objet (OO). Le Component Object Model (COM) n'établit un lien entre ses composantes que si elles résident toutes sur le même ordinateur. COM a par la suite été élargi en Distributed Component Object Model (DCOM). DCOM est un COM avec l'addition d'un protocole de communication réseau qui permet l'interopérabilité entre des SI résidants sur des ordinateurs différents <sup>14</sup>.

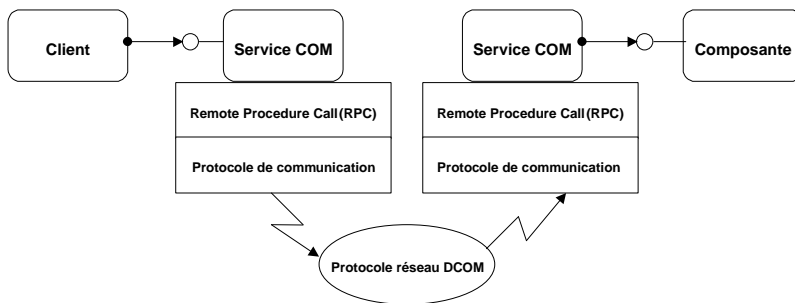


Figure 2-6 Le protocole de DCOM (adapté, traduit et redessiné) <sup>15</sup>

L'exécution d'une composante DCOM se fait où cela fait le plus de sens. Plusieurs composantes peuvent être impliquées dans une même transaction, et leur exécution se fera tantôt sur l'ordinateur client dans le cas de l'interface utilisateur par exemple, tantôt sur l'ordinateur serveur, dans le cas de la base de données distante.

DCOM peut être implémenté dans n'importe quel langage de programmation qui supporte les services COM. Il est donc indépendant de tout langage.

Finalement, DCOM étant un produit Microsoft, il fonctionne surtout sur les systèmes d'exploitation Windows, mais est aussi disponible pour UNIX et Linux avec un certain succès.

<sup>14</sup> Microsoft Corporation, "DCOM Technical Overview",  
<[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn\\_dcomtec.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomtec.asp)>,  
Novembre 1996

<sup>15</sup> Idem



## 2.5 CORBA

« Le Common Object Request Broker Architecture ou CORBA est une technologie d'interopérabilité orienté objet définie par le Object Management Group »<sup>16</sup>. C'est une technologie indépendante des langages de programmation et des systèmes d'exploitation. Son IDL a été « choisi par la "International Standards Organization" (ISO) comme un langage universel pour décrire les interfaces aux composantes logicielles (norme ISO/IEC DIS 14750) »<sup>17</sup>.

Des objets écrits dans des langages de programmation différents peuvent s'invoquer. L'Object Request Broker (ORB) agit comme un bus de communication.

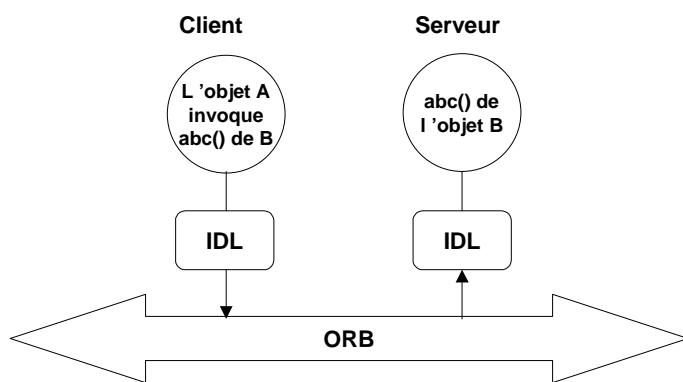


Figure 2-7 Invocation CORBA (traduit et redessiné)<sup>18</sup>

CORBA est entièrement supporté par tous les systèmes d'exploitation car il n'est qu'une spécification à être implémentée.

<sup>16</sup> DISA/JIEO, "DII COE Distributed Applications Series - Recommendations for using DCE, DCOM, and CORBA Middleware", 13 avril 1998

<sup>17</sup> Computer Reseller News, "MOG pushes to become standards body, touts CORBA", <<http://www.techweb.com/se/directlink.cgi?CRN19970728S0027>>, juillet 1997

<sup>18</sup> Segue Software, "CORBA Primer", <<http://cgi.omg.org/library/seguecorba.pdf>>, 2002

## 2.6 Java et RMI

Geopalan Suresh Raj explique le fonctionnement du RMI de Java : « Le Remote Method Invocation ou RMI est lié à un protocole appelé Java Remote Method Protocol (JRMP). Les objets sont transmis à l'aide de ce protocole et il est donc obligatoire que le client et le serveur soient tous deux écrits en langage Java. Chaque objet définit une interface pour l'accès des méthodes. Un registre appelé RMIRegistry est exécuté sur le côté serveur et contient l'information sur les objets serveurs disponibles »(traduction libre)<sup>19</sup>

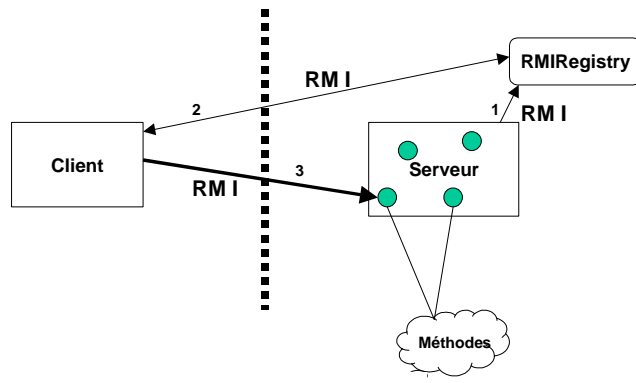


Figure 2-8 Le Remote Method Invocation(traduit et redessiné)<sup>20</sup>

Le RMI peut être utilisé sur n'importe quel système d'exploitation qui supporte la Machine Virtuelle Java (JVM). Parce que tous les objets impliqués dans un échange RMI sont tous des objets Java, le processus est simple et efficace.

En addition au protocole JRMP pour le RMI, Java supporte le IIOP de CORBA et implémente CORBA et DCOM afin d'échanger des données avec des applications d'autres langages que Java.

Le prochain chapitre parlera de l'interopérabilité s'appliquant au XML.

<sup>19</sup> Raj, Geopalan Suresh, "A Detailed Comparaison of CORBA, DCOM and Java/RMI", mai 2002, Copyright © 1997-2001 Gopalan Suresh Raj, 8730, S. Country Drive, Apt #201, Oak Creek, WI 53154-3880 U.S.A. All rights reserved.

<sup>20</sup> Sun Microsystems, "An Overview of RMI Applications", <<http://java.sun.com/docs/books/tutorial/rmi/overview.html>>, 18 mai 2002, Copyright 1995-2002 Sun Microsystems, Inc. All rights reserved.

## Chapitre 3 - Le XML et ses composantes

### 3.0 Le XML – Un historique

XML signifie : Langage de Balisage Étendu ou eXtensible Markup Language. C'est un standard récent (adopté en 1998 par le W3C) qui s'appuie sur un standard solide, le Standard Generalized Markup Language (SGML) dont les origines remontent à 1960.

« XML est un langage permettant d'inclure des données structurées dans un fichier texte »<sup>1</sup>. XML permet de séparer le contenu de la présentation.

XML est considéré comme un "méta langage": c'est à dire qu'il permet de définir d'autres langages. Pour cette raison on dit que c'est un langage orienté métier, d'ou les vocabulaires structurés comme le MathML (Mathématiques), CML(Chimie), MilML (Militaire) qui seront traités dans les chapitres suivants.

La force de XML réside dans sa capacité à décrire n'importe quel domaine de données grâce à son extensibilité. Il permet de structurer, poser le vocabulaire et la syntaxe des données qu'il contient.

#### 3.0.1 SGML

« Le SGML est le résultat des travaux de trois chercheurs d'IBM qui devaient trouver une solution pour traiter des milliers de documents légaux créés sur différents systèmes dont chacun utilisait un format propriétaire. Les trois chercheurs créèrent GML (Generalized Markup Language), un langage permettant de décrire un format de document. Le GML évolua et en 1968 un groupe de développeurs internationaux adoptèrent le SGML (Standard Generalized Markup Language) qu'ISO approuva en 1986 »<sup>2</sup>.

#### 3.0.2 HTML

« Berners-Lee, l'inventeur du Web, désirait une façon simple de présenter ses documents sur le Web et il créa le HTML (HyperText Markup Language), qui est plus limité et non extensible que le SGML. La force du HTML est la présentation visuelle d'informations, il n'a aucune manière de structurer les données contenues dans l'information »<sup>3</sup>.

Le HTML utilise lui aussi des balises. Cependant, les balises du HTML font partie d'un vocabulaire limité et ne donnent que des instructions à un navigateur internet sur la façon d'afficher l'information qui se trouvent entre ses balises.

---

<sup>1</sup> Giguère, Charles, "Comprendre XML", <<http://membres.lycos.fr/cgiguere/vdn/vdn85.htm>>, 2001.

<sup>2</sup> Idem

<sup>3</sup> Idem

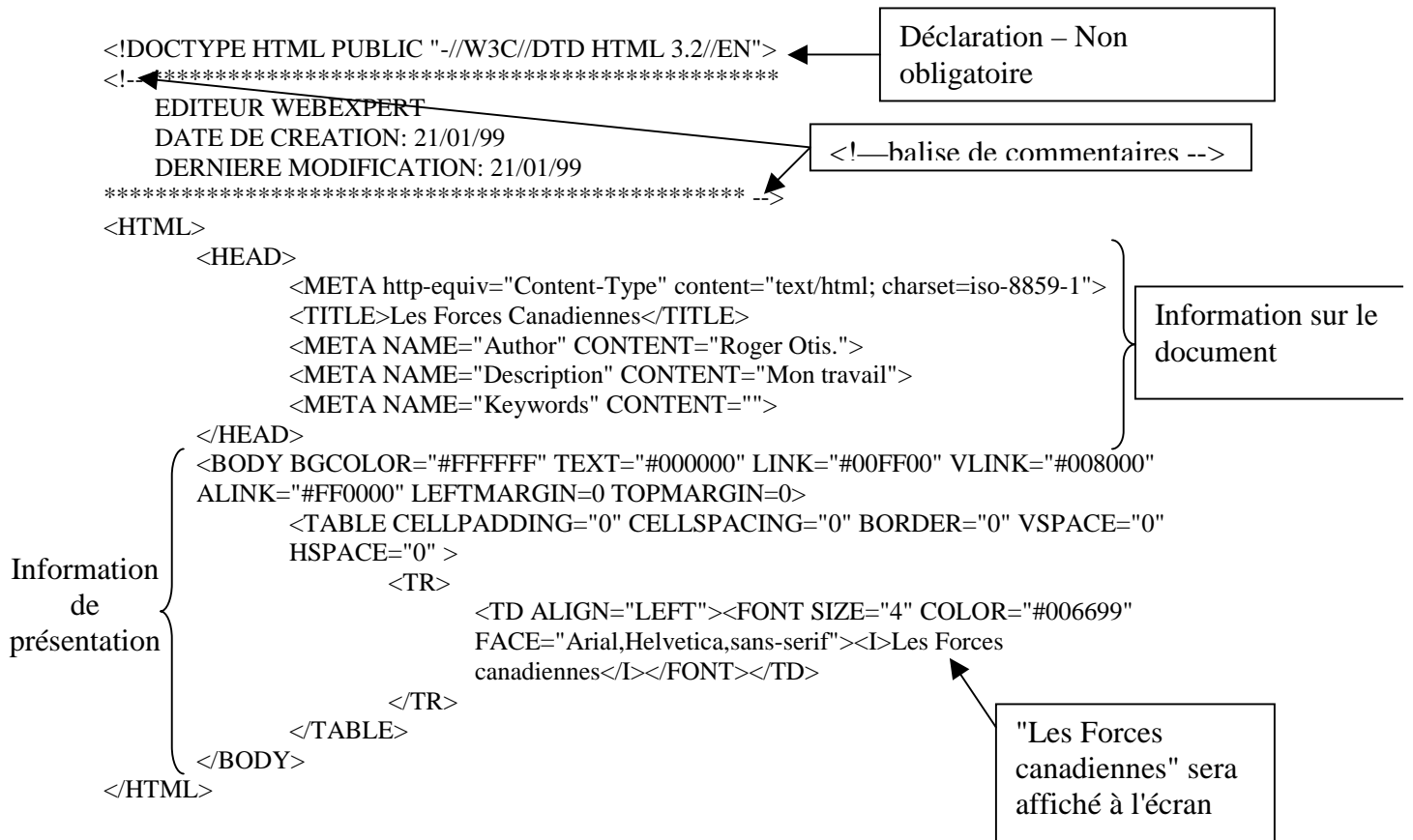
### 3.0.3 Le World Wide Web Consortium (W3C)

Vers 1996, le W3C initia le développement d'un langage de structuration des données : XML. Le standard XML fût adopté officiellement en 1998. Le groupe de travail sur le XML du W3C est formé principalement de membres de l'industrie de l'informatique, des télécommunications, du multi-média, d'organismes gouvernementaux et militaires, venant de partout au monde.

### 3.1 Introduction au XML

Le XML est un langage qui utilise des balises. On dit qu'il est extensible en raison de sa capacité à créer de nouvelles balises. Un document XML est donc mis en forme à l'aide de balises, comme le HTML en quelque sorte.

La différence principale entre les balises du HTML et celles du XML est dans la signification. Alors que les balises du HTML décrivent la manière de présenter l'information, les balises du XML décrivent l'information elle-même. Voyons un exemple pour bien comprendre:



L'exemple HTML que l'on vient de voir a pour but d'afficher "Les Forces canadiennes" de la façon suivante:

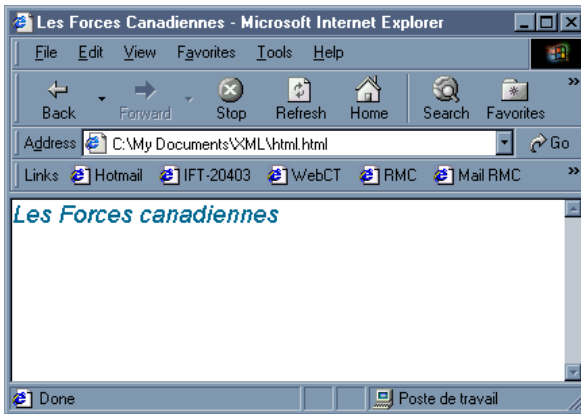
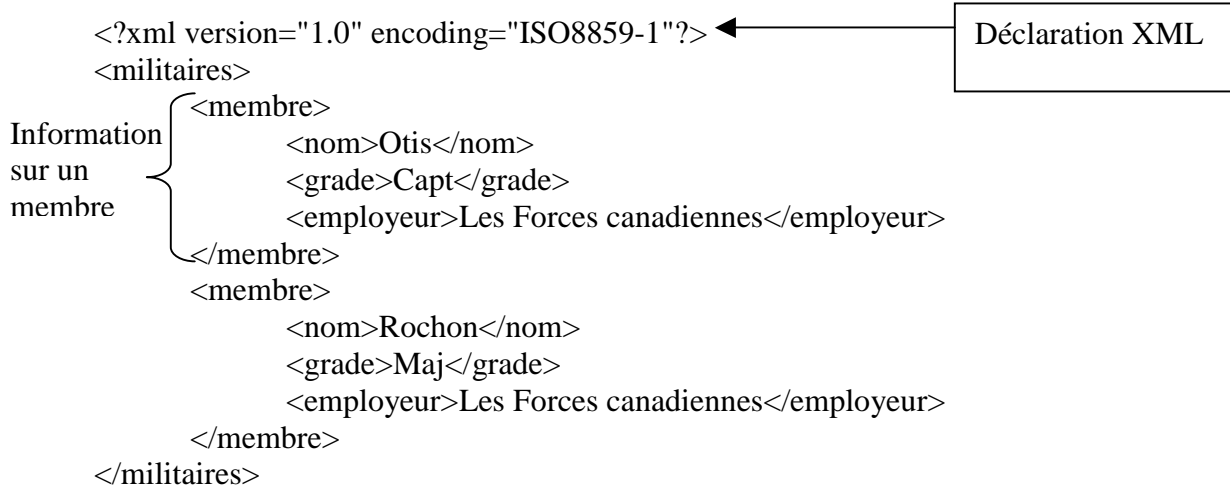


Figure 3-1 Affichage d'un document HTML

On voit bien que les balises HTML donnent l'information au navigateur au sujet de la grosseur des caractères, de la couleur du texte, de sa mise en forme, de sa justification, etc...

Prenons l'information véhiculée dans l'exemple précédent et voyons un exemple XML.



Ici, les balises XML identifient l'information qui est stockée dans le document XML et ne donnent aucune instruction d'affichage ou de présentation. Il n'est pas permis en HTML de créer ses propres balises comme nous l'avons fait ici avec les balises XML. C'est ce qui a permis d'identifier notre information.

Les navigateurs internet sont tout de même capables d'afficher notre information selon le même format que celui utilisé lors de la rédaction du code. Pour afficher l'information d'une autre manière, il faut passer par un autre langage de présentation comme "Cascading StyleSheet" (CSS), "eXtensible StyleSheet Language" (XSL) ou "eXtensible StyleSheet Language Transformation" (XSLT).

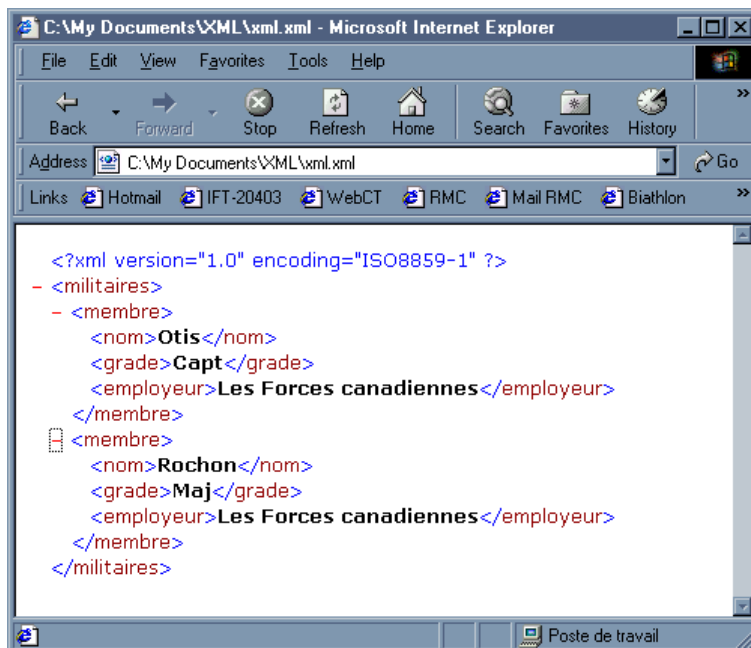


Figure 3-2 Affichage de base d'un document XML

Ce qui est donc important dans un document XML, c'est l'information qu'il contient, un peu comme une base de données. Les balises peuvent être considérées comme des noms de champs. Cela constitue un avantage primordial dans l'échange de données, en autant que l'application destination est en mesure de lire le document XML. Nous reviendrons sur cet aspect plus loin.

Le XML possède des qualités appréciables comme support d'information par rapport à des moyens plus complexes et plus lourd comme des bases de données, toujours dans un contexte d'échange d'information.

Le XML est **lisible** autant par un ordinateur (à l'aide d'un analyseur XML, ou parseur) que par un humain. Vous n'avez certainement pas eu de difficultés à lire la figure 3-2 et surtout à comprendre la signification de l'information qui y est contenue. Aucune connaissance spécifique n'est requise pour lire un document XML.

Le XML est **extensible** en permettant d'ajouter des balises pour décrire avec précision des éléments propre à un domaine spécifique (le domaine militaire par exemple).

Le XML est facilement **transportable** par sa nature de fichier texte. Tout protocole à même de transporter du texte peut transporter un document XML et l'information qu'il contient.

### 3.1.1 La syntaxe du XML

Le W3C a établi une norme précise pour la formation de documents XML (Norme XML 1.0). Le XML possède une syntaxe stricte et robuste qui en font un langage normalisé qui facilite son utilisation dans un contexte d'interopérabilité.

Un document XML est constitué de trois parties:

- a. la déclaration XML;
- b. la déclaration de type de document; et
- c. l'arbre des éléments.

**Déclaration XML.** Tout document XML débute par une déclaration, qui constitue un type d'en-tête de document. La déclaration précise la version du XML qui est utilisée.

**Déclaration de type de document.** On y retrouve une déclaration qui précise le modèle sur lequel se base le document XML. C'est une référence à un "Document Type Definition" (DTD) ou à un schema XML. Nous verrons plus loin ces deux types de documents.

**L'arbre des éléments.** C'est là que se retrouve le contenu du document XML. Les données sont organisées sous forme arborescente ou hiérarchique, à l'intérieur de balises, ce qui permet un traitement aisé de l'information par un analyseur XML qui sera chargé d'extraire et de traiter l'information du document XML.

Le XML comporte aussi des règles, telle une grammaire, afin de structurer adéquatement le document. Ces règles constituent la norme XML.

**Balise.** C'est la base du XML. Une balise est une indication dans le document qui a une signification précise et un format précis. La balise habille en quelque sorte l'information qu'on introduit dans le document. Une balise est une chaîne de caractères qui à la forme suivante:

```
<organisation>
```

Pour toute balise dite d'ouverture, comme dans cet exemple, il doit y avoir une balise de fermeture qui prend la forme suivante et entre lesquelles l'information se retrouvera:

```
</organisation>
```

Le HTML permet d'omettre certaines balises de fermeture comme la balise de saut de paragraphe </P>. Le XML oblige la balise de fermeture dans toutes les situations. Cette obligation augmente la lisibilité et la compréhension du document. La délimitation des éléments du document est assurée par cette contrainte. Il ne sera donc pas possible de méprendre un élément pour un autre si toutes les balises sont bien en place.

Tout ce qui se trouve entre deux balises d'ouverture et de fermeture fait partie de cet ensemble.

```
<organisation name="brigade">
  <unite>
    <UIC>1735</UIC>
    <nom>5e RALC</nom>
  </unite>
  <unite>
    <UIC>1254</UIC>
    <nom>12e RBC</nom>
  </unite>
  <unite>
    <UIC>3540</UIC>
    <nom>2eR22eR</nom>
  </unite>
</organisation>
```

**Attribut.** L'exemple précédent introduisait la notion d'attribut avec l'utilisation de name="brigade" à l'intérieur de la balise <organisation>, ce qui donne:

```
<organisation name="brigade">
```

Cet attribut précise à quel type d'organisation on réfère. Toute balise peut avoir un ou plusieurs attributs. Ils complètent l'information qui est à l'intérieur des balises.



## 3.2 Les composantes du XML

### 3.2.1 DTD

Chaque document XML doit être validé afin de s'assurer qu'il respecte la syntaxe du domaine spécifique auquel il se rapporte, sinon il sera déclaré document non-valide. Pour accomplir cette tâche, on a recours entre autre aux Document Type Definition (DTD). Un DTD est un document qui spécifie des règles qui dictent la structure du document XML à écrire. Un DTD décrira par exemple que certains éléments ne peuvent avoir qu'un seul élément subordonné.

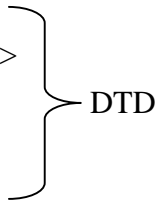
Pour être valide, un document XML doit déclarer le DTD qui le gouverne. De plus, il doit se conformer aux règles énoncées dans le DTD.

#### 3.2.1.1 Définition de DTD

Un DTD peut être défini de 2 façons:

- a. **Forme interne.** Le DTD est inclus dans le document XML:

```
<?xml version="1.0" encoding="ISO8859-1"?>
<! ELEMENT militaire (membre)>
<! ELEMENT membre (nom, grade, employeur)>
<! ELEMENT nom #PCDATA >
<! ELEMENT grade #PCDATA >
<! ELEMENT employeur #PCDATA >
<militaires>
  <membre>
    <nom>Otis</nom>
    <grade>Capt</grade>
    <employeur>Les Forces canadiennes</employeur>
  </membre>
  <membre>
    <nom>Rochon</nom>
    <grade>Maj</grade>
    <employeur>Les Forces canadiennes</employeur>
  </membre>
</militaires>
```



- b. **Forme externe.** Une déclaration dans le fichier XML mentionne l'emplacement du DTD. Cela peut être le chemin d'accès d'un fichier sur le disque dur de l'ordinateur ou une adresse URL. Utiliser SYSTEM pour une adresse URL et PUBLIC pour un chemin d'accès.

```
<?xml version="1.0" encoding="ISO8859-1"?>
<!DOCTYPE militaire SYSTEM http://www.siteweb.com/DTDs/militaire.dtd >
<militaires>
  <membre>
    <nom>Otis</nom>
    <grade>Capt</grade>
    <employeur>Les Forces canadiennes</employeur>
  </membre>
  <membre>
    <nom>Rochon</nom>
    <grade>Maj</grade>
    <employeur>Les Forces canadiennes</employeur>
  </membre>
</militaires>
```

ou

```
<?xml version="1.0" encoding="ISO8859-1"?>
<!DOCTYPE militaire PUBLIC c:/xml/DTDs/militaire.dtd >
<militaires>
  <membre>
    <nom>Otis</nom>
    <grade>Capt</grade>
    <employeur>Les Forces canadiennes</employeur>
  </membre>
  <membre>
    <nom>Rochon</nom>
    <grade>Maj</grade>
    <employeur>Les Forces canadiennes</employeur>
  </membre>
</militaires>
```

### 3.2.2 Les Schémas XML

Les DTD ne sont pas écrits en langage XML. Ils dérivent plutôt du plus ancien SGML. Il en résulte que les analyseurs XML ne peuvent interpréter un DTD. La popularité de l'échange de données grâce au XML amène avec elle la notion très importante de type de données (caractère, entier, réel, date, etc...). Le DTD ne peut définir qu'un seul type de données, soit le caractère/chaîne de caractère, ce qui limite son champ d'application. Le traitement de données numériques doit se faire par programmation en utilisant des DTD.

Le W3C a donc introduit la notion de schéma XML. Le but du schéma, comme un DTD est de formuler des règles pour la formation d'un document XML. De plus, le schéma peut traiter différents types de données et même créer des types de données complexes pour satisfaire des besoins précis. Le schéma XML met à la disposition de l'échange de données une flexibilité adjointe d'une grande rigueur de structuration de document. Le schéma XML est un document XML valide.

Nous verrons aussi l'utilisation des espaces de noms avec les schémas XML, qui donnent un contexte au vocabulaire utilisé.

Un schéma XML débute par une déclaration XML. Les déclarations d'éléments et d'attributs se font à l'intérieur de la balise `<xs:schéma>`. Tout élément qui appartient au schéma doit être précédé par `xs:`. Voici un exemple de schéma XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs=http://www.w3.org/2000/10/XMLSchema>
  <xs:element name="militaires">
    <xs:complexType>
      <xs:sequence>

        <xs:element name="membre">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nom" type="string"/>
              <xs:element name="grade" type="string"/>
              <xs:element name="employeur" type="string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>

      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### 3.2.2.1 Les éléments et attributs d'un schéma XML

Un document XML est composé d'éléments qui sont les balises du document. Comme vu précédemment dans la section 3.1.1., les éléments peuvent comporter des attributs. L'exemple précédent déclarait plusieurs éléments de la façon suivante à l'intérieur de la balise <xs:schema>:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xsd=http://www.w3.org/2000/10/XMLSchema>
.
.
.
    <xs:element name="nom" type="string"/>
    <xs:element name="grade" type="string"/>
    <xs:element name="employeur" type="string"/>
.
.
.
</xs:schema>
```

Nous avons vu qu'un élément d'un document XML pouvait contenir des attributs à l'intérieur de sa balise:

```
<organisation name="brigade">
```

Comme un schéma XML est un document XML, ses éléments peuvent aussi contenir des attributs. Ainsi, les éléments "nom", "grade" et "employeur" ont tous un attribut "type" qui indique le type de données qui sera utilisé par le document XML que ce schéma définit.

### 3.2.2.2 Les Types de données

Les schémas XML permettent le typage des données, ce qui n'était pas le cas avec les DTD. Les schémas donnent ainsi aux documents XML une plus grande capacité d'échange de données et permet même de déclarer ses propres types de données. Il y a trois catégories de types de données qui concernent les schémas XML:

- a. type simple;
- b. type complexe; et
- c. type listes.

#### 3.2.2.2.1 Les types simples

Les schémas XML utilisent une bibliothèque de types de données prédéfinies.

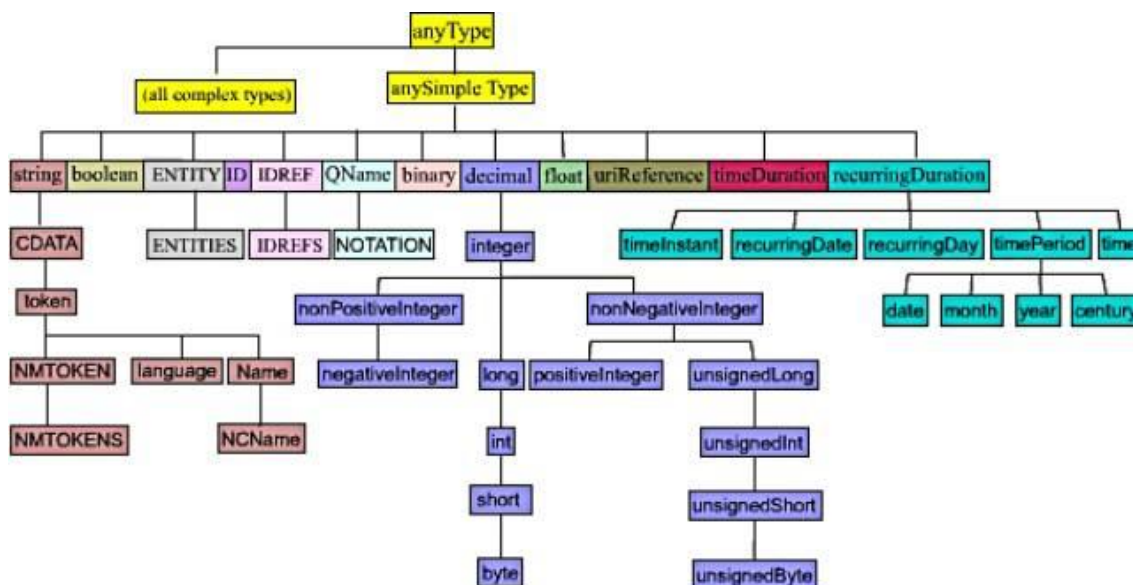


Figure 3-3 Types de données prédéfinis de XML Schéma <sup>4</sup>

On retrouve dans cette bibliothèque les types de données simples les plus courants: chaînes de caractères, entiers, dates, réels etc. Aussi, le type de données CDATA (caractère/chaîne de caractère) utilisé par les DTD est supporté par les schémas XML.

L'utilisation de ces types de données par les schémas et les documents XML permet une validation automatique des types de données transportés à l'intérieur des balises et l'insertion dans une base de données sans autre forme de validation. Cet aspect

<sup>4</sup> Chazalon, Grégory et Lemoine, Joséphine, "Les Schémas XML", <<http://zuse.esnig.cifom.ch/intranet/SpecSchema/xschema.htm>>, 21 janvier 2001

du schéma XML sauve du temps de développement en programmation des applications d'interopérabilité.

### 3.2.2.2.2 Les types complexes

Un autre avantage exceptionnel du schéma XML par rapport au DTD est sa capacité à créer des types de données complexes. Le concepteur d'un schéma XML peut décider de la structure d'un nouveau type de données. Cette capacité s'apparente à l'instruction "typedef" du langage C et donne beaucoup de flexibilité.

Dans notre exemple de schéma XML, nous avons défini deux types de données complexes, soit le type de données "militaires" et le type de données "membre".

```
<xs:element name="militaires">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="membre">
        <xs:complexType>
          <xs:sequence>
            .
            .
            .
```

Cette caractéristique de définition de type de données se nomme encapsulation et est très utilisé en informatique. Ainsi le type complexe militaires encapsule le type complexe membre et à son tour le type complexe membre encapsule les éléments "nom", "grade" et "employeur". C'est ce qui permettra au document XML qui sera validé par ce schéma d'utiliser la hiérarchie suivante: militaire → membre → nom.

### 3.2.2.2.3 Les types listes

XML Schéma possède trois types de listes intégrés. Il est aussi possible de créer des types listes à partir de types existants. On peut utiliser le type liste dans un rôle plus spécifique que le type qui le compose dans un document XML.

### 3.2.2.3 Les espaces de nommage ou "namespaces"

Les espaces de nommage fournissent un contexte à un vocabulaire, ce qui facilite la création de schémas et la validation de documents. Pour utiliser un espace de nommage avec un DTD, il faut rendre le DTD complexe et très peu lisible, en empruntant des détours. Le schéma XML permet d'intégrer aisément les espaces de nommage et donne encore plus de capacités et de flexibilité à nos documents XML.

Un espace de nommage définit l'espace où un élément spécifique sera utilisé et où il a une signification. Si deux objets portent le même nom, il n'y a pas de conflit tant qu'ils n'occupent pas le même espace.

Si une seule personne dans un espace donné se nomme Michel, dire "Bonjour Michel" est tout à fait compréhensible. Si deux personnes dans ce même espace se nomment Michel, il faut ajouter le nom de famille ou utiliser un surnom.

Le but de l'espace de nommage est donc de différencier deux éléments ayant le même nom afin de garder l'intégrité référentielle de l'information.

Le schéma XML utilise un "Uniform Resource Identifier" (URI) pour identifier de manière unique un espace de nommage. Les éléments ou attributs qui utilisent un espace de nommage y sont associé à l'aide d'un préfixe:

```
<?xml version="1.0"?>
  <mil:armee xmlns:mil="http://www.exemple.com/militaire">
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <mil:element name="armee">
        <xs:element name="militaire">
          .
          .
          .
        </xs:element>
      </mil:element>
    </xs:schema>
  </mil:armee>
```

Tout élément ou attribut placé entre les balises <mil:armee> sera considéré par un analyseur XML comme faisant partie de cet espace. Par exemple, nos éléments "nom", "grade" et "employeur" seront identifiés comme tel.

Un autre système différent utilisant lui aussi un nom d'élément "nom" par exemple n'entrera pas en conflit avec le nôtre car il n'utilise pas le même espace de nommage. Ceci constitue une nette amélioration par rapport au DTD.

Par exemple, une balise canadienne avec espace de nommage comportant de l'information sur les opérations <cal :ops></cal :ops> n'entrera pas en conflit avec le

même type d'information venant d'un document XML américain et utilisant l'espace de nommage `<us :ops></us :ops>`.

### 3.2.3 Document Object Model (DOM) et Simple API for XML (SAX)<sup>5</sup>

La récupération des données se trouvant dans un document XML s'effectue à l'aide d'un analyseur XML. L'analyseur parcourt le document XML et en extrait les informations.

Il existe deux types majeurs d'analyseur XML selon la technique qu'ils utilisent :

- a. **Hierarchique/arborescente:** Construction en mémoire d'une structure d'arbre hiérarchique contenant des objets représentant les éléments du document. Le Document Object Model (DOM) utilise cette méthode; et
- b. **Événementiel:** Réagit à des événements comme le début ou la fin d'un élément et retourne le résultat à l'application. L'API. SAX (Simple API for XML) utilise cette technique.

#### 3.2.3.1 Comparaison entre DOM et SAX

Les analyseurs DOM doivent construire un arbre des éléments du document XML en mémoire. Si le document XML est imposant, cela peut causer des problèmes de ressources. Il peut aussi être lent d'analyser un gros document XML pour la même raison.

La gestion de la mémoire est plus efficace avec un analyseur de type SAX car on n'a pas à utiliser le double de ressource mémoire. Par contre, SAX recherche des événements. Il ne retourne pas en arrière comme peut le faire DOM en parcourant de haut en bas son arbre.

Donc selon l'utilisation des documents XML, leur fréquence de consultation et les efforts de recherche y étant faits, un type d'analyseur spécifique sera utilisé.

---

<sup>5</sup> Pillou, Jean-François, "Parser du XML – Les API DOM et SAX", <http://www.commentcamarche.net/xml/xmldomsax.php3> >, 2001.



### **3.3 L'interopérabilité et le XML**

Le chapitre deux parlait des différentes technologies d'interopérabilité, soit COM/DCOM, CORBA, et le RMI de Java. Toutes ces technologies peuvent implémenter l'interopérabilité dans un SI. Nous avons vu comment ces technologies pouvaient cependant être complexes à implanter et les connaissances requises pour le faire.

Nous avons aussi parlé de la norme HTTP comme dominant les protocoles de communication sur internet de par sa simplicité et son économie d'exécution.

Le XML pourrait bien être ce plus petit dénominateur commun qui permet un échange d'information, car le XML est simple, basé sur des fichiers textes, peut contenir tous les types de données, peut être parcouru et compris par tous les analyseurs XML, sur n'importe quelle plate-forme, en utilisant le protocole de communication le plus efficace.

Le XML est non seulement lisible par les ordinateurs, mais aussi par l'humain. Il définit le vocabulaire du secteur d'activité qu'il supporte et peut même s'accommoder de vocabulaires différents, pour autant qu'il sache comment adapter l'un à l'autre.

#### **3.3.1 La plate-forme XML**

La puissance du XML ne tient pas au fait qu'il peut emmagasiner de manière plus efficace des données (ce qui n'est pas le cas), mais parce qu'il est la première technologie d'interopérabilité à prendre en charge les données complexes de manière simple. Il s'occupe de l'information et de ses relations avec d'autres types d'information, de manière dynamique.

L'information est souvent irrégulière et hiérarchique. Les relations père-fils entre différentes informations peuvent être efficacement traitées par le XML.

La meta-information (information à propos de l'information) est essentielle afin d'expliquer le type de données auxquelles ont fait face, mais aussi le contexte et la signification de l'information qu'on retrouve dans la base de données. Le XML est totalement basé sur la meta-information.

Finalement, ce que la plate-forme XML apporte est l'utilisation d'un format de fichier universel en ce sens que tous les systèmes d'exploitation, tous les langages de programmation, peuvent recevoir et lire un document XML

### 3.3.2 Simple Object Access Protocol (SOAP)

SOAP est un protocole de communication dont la fonction est l'échange d'information entre des ordinateurs peu importe leur système d'exploitation ou l'environnement de programmation. SOAP peut être utilisé autant sur internet que sur un réseau local (LAN), dans un environnement décentralisé et distribué.

SOAP est structuré selon la norme XML. Il en a donc toutes les caractéristiques et en tire tous les bénéfices. SOAP définit un mécanisme simple pour échanger des messages en utilisant des enveloppes (écrites en XML) dans lequel il insère son modèle modulaire. Ainsi SOAP peut envoyer une requête et la façon d'y répondre, selon le principe du "Remote Procedure Call" (RPC).

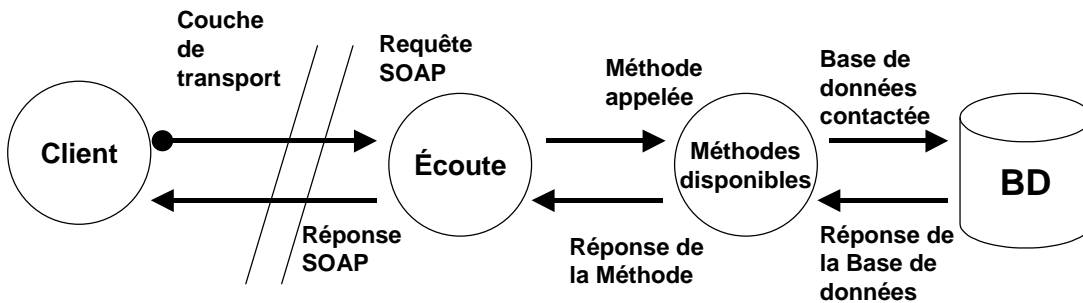


Figure 3-4 L'appel SOAP (redessiné) <sup>6</sup>

Voici la structure d'un message SOAP:

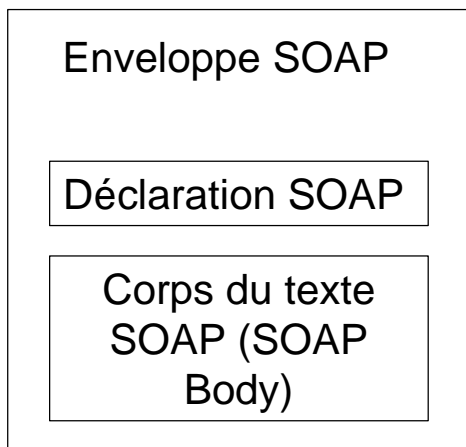


Figure 3-5 Structure d'un message SOAP (traduit et redessiné) <sup>7</sup>

<sup>6</sup> Soapuser.com, "Bases SOAP – Qu'est-ce que SOAP?", < <http://www.soapuser.com/fr/basics1.html> >, 2001

Voici de quoi à l'air un message SOAP:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelop
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop/"
  SOAP-ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>
  <SOAP-ENV:Body>
    <m:GetHighestRank xmlns:m="http://www.exemple.com/methodes/"
    </m:GetHighestRank>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelop>
```

Ce message SOAP, lorsque reçu par un serveur qui écoute l'arrivée de message SOAP, fait appel à une méthode `GetHighestRank` définie à l'URL fourni et qui demande un retour au client de la réponse.

Ce genre de message peut être envoyé à partir de n'importe quel système d'exploitation, disons Windows NT vers n'importe quel autre, disons UNIX. Ce message fait abstraction des protocoles de communication, car il ne précise pas quel protocole utiliser. Il est transporté par n'importe quel protocole disponible capable de transporter du texte.

SOAP est une alternative à CORBA, DCOM et RMI. Il est beaucoup plus simple à utiliser, il est facilement lisible tant par l'humain que par l'ordinateur, et ne demande que très peu de support pour son installation. Il est moins performant dans certaines circonstances que ces rivaux comme CORBA, mais sa flexibilité et son universalité en font un sérieux compétiteur dans le domaine d'échange d'information.

Dans un contexte d'interopérabilité au niveau de l'OTAN, l'utilisation de SOAP versus d'autres technologies permettrait une efficacité et une flexibilité accrues en raison de la simplicité d'utilisation de SOAP. L'expertise requise pour l'implanter rapidement dans des conditions difficiles est moindre.

---

<sup>7</sup> Birbeck, Mark et autres, "Professional XML 2<sup>nd</sup> Edition", 2001

## Chapitre 4 – Sécurité entourant le XML

### 4.0 Généralités

On parle fréquemment d'un risque relié à la sécurité dans l'utilisation du XML car on interprète sa facilité d'utilisation comme une vulnérabilité contre les attaques mal intentionnées. Ce sujet d'importance pour les applications « business to business » (B2B) a été débattu longuement et des solutions viables se raffermissent. Ces mêmes solutions sécuritaires s'appliqueront dans la poursuite de l'interopérabilité des forces militaires pour des systèmes à accès plus restreints, tel un éventuel intranet pour l'OTAN. Dans un premier temps, suite à l'identification de la menace, on présentera des outils XML sécurisant le transfert d'information XML entre deux applications. Ensuite on discutera du potentiel prometteur de l'utilisation des technologies XML pour accroître la sécurité dans un éventail d'applications réseau.

### 4.1 Protection d'instances d'utilisation du XML

Il est très facile de manipuler et modifier un document XML dans son état original ce qui le rend vulnérable. En consultant la figure 4-1, on notera les instances dans lesquels un document pourrait être compromis soit durant son utilisation, lorsqu'il est sauvegardé en mémoire ou en empruntant certains canaux de transmission. Ces instances constituent la menace. Ainsi du point de vue du destinataire, le message XML reçu ne représente pas une source d'information fiable comprenant les caractéristiques de sécurisation souhaitable selon les critères suivant :

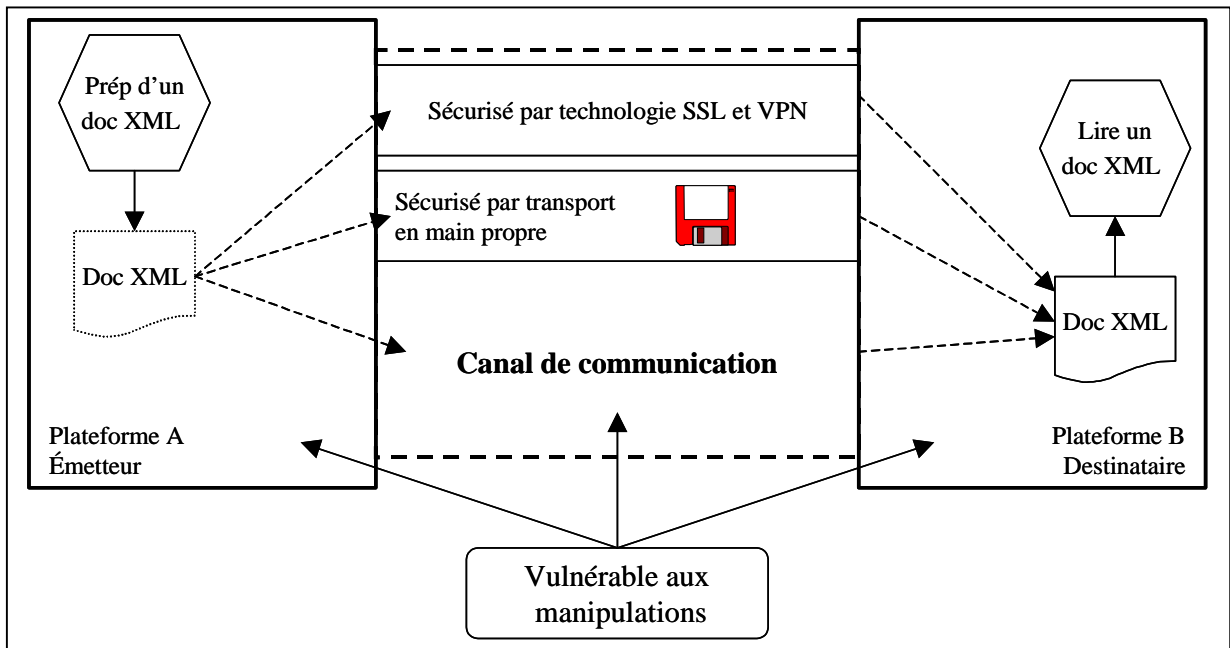


Figure 4-1 Vulnérabilité des documents XML échangés

- a. Manque d'indication de l'efficacité du contrôle d'accès système<sup>1</sup>. Le contrôle d'accès système est la barrière physique et virtuelle qui contrôle le périmètre d'une plateforme permettant à un utilisateur de jouir des services d'un serveur. En mode réseau on y inclut les réseaux privés virtuels (VPN), les barrières antivirus, les firewalls et l'accès à distance;
- b. L'authentification n'est pas assurée. L'authentification c'est l'établissement de l'identité de l'émetteur ou du destinataire<sup>2</sup>;
- c. La confidentialité n'est pas assurée. On atteint la confidentialité d'une transmission lorsqu'on est certain que seul le destinataire a été capable de lire un message<sup>3</sup>;
- d. L'intégrité n'est pas assurée. Le message maintient son intégrité si le destinataire reçoit exactement tout le message envoyé par l'émetteur<sup>4</sup>; et
- e. La non répudiation ne pourra être invoquée plus tard. La non répudiation c'est l'authenticité de partenaires qui se sont engagés dans une transaction pour laquelle aucun d'eux ne peut nier y avoir participé.

Il y a des solutions courantes telles que Secure Socket Layer (SSL) et le Virtual Private Network (VPN) qui protègent l'ensemble d'un document durant sa transmission mais pas les données<sup>5</sup>. Le destinataire ne peut toujours pas être certain de la nature du message qu'il aura reçu. De même, le problème est inverse pour l'émetteur qui attend une réponse du destinataire. Il n'aura aucun moyen de s'assurer de l'authentification, de la confidentialité et de l'intégrité du message de retour.

## 4.2 XML Key Management Specification (XKMS)

En réponse à ces risques et pour parer à la souplesse et à la nature éphémère des données XML, des outils de sécurité XML prennent forme sous la tutelle du W3C. La pierre angulaire de ce système de sécurité se nomme « XML Key Management Specification (XKMS) ». XKMS définit une architecture d'enregistrement et de distribution de clefs publiques de chiffrement. XKMS pourra remplacer les formats protocoles « Public Key Infrastructure » (PKI) (PKIX, Card Management Services, Online Certificate Status Protocol (OCSP)) par des documents XML transportés par

---

<sup>1</sup> OCTO Technology, Le Livre Blanc de la Sécurité, septembre 2001, page 7, ([www.octo.com/fr/techno/download/wp-secu.pdf](http://www.octo.com/fr/techno/download/wp-secu.pdf), 5 mai 2002).

<sup>2</sup> Idem, page 10.

<sup>3</sup> Idem, page 29.

<sup>4</sup> Idem.

<sup>5</sup> InternetWeek.com, Securing XML At The Data Level, (<http://www.internetwk.com/story/INW20020503S0002>, 3 mai 2002).

SOAP. XKMS s'appuie de deux spécifications à savoir le XML Signature et le XML Encryption<sup>6</sup>. Ces spécifications méritent plus d'attention.

#### 4.2.1 XML Signature

La mise en place du XML Signature prévient la falsification d'un message causée par un intermédiaire représenté à la figure 4-2. Le récepteur est certain de l'intégrité, de la non répudiation et de l'authentification d'un message dès qu'il en a vérifié la signature. Le document reçu peut être retourné ou envoyé à un nouveau destinataire. L'identité de l'émetteur est indépendante du terminal utilisé car c'est l'individu qui est identifié. La signature est possible sur n'importe quelle portion d'un document électronique polyvalent qui peut au besoin être retransmis à plusieurs signataires.

- a. L'information d'un document XML signé contient<sup>7</sup>: la syntaxe XML pré-définie selon la méthode de canonicalisation (XML-C14N), l'algorithme de la signature, l'algorithme utilisé par le « digest », de l'information de référence sur les données signées, des caractéristiques optionnelles sur le contexte de la signature (date-heure, no. de série), un manifeste optionnel comprenant des autorisations regroupées d'un ou de plusieurs utilisateurs;
- b. La génération de la signature d'une instance XML commence par la canonicalisation allégeant la grosseur des données suivit par le processus normal de signature électronique utilisant un digest. Le digest est signé avec la clé privée de l'auteur; et
- c. Le récepteur du message vérifie la signature en deux étapes. Tout d'abord, l'authentification et la non répudiation de la signature est effectuée. Par la suite, on vérifie la valeur du digest pour s'assurer que le contenu du message n'a pas été modifié.

#### 4.2.2 XML Encryption

L'outil rendant le service de confidentialité est la spécification XML Encryption ajouté en représentation à la figure 4-2. Le chiffrement est crucial pour la sécurité des données XML surtout si elles sont transmises par des circuits non protégés. La particularité du XML Encryption permettra à différents destinataires de lire différentes parties d'un document. Toutefois, on se doit de maintenir la sécurité à chaque nœud.

---

<sup>6</sup> JDNet Solutions, Le W3C complète le chapitre de la sécurité des services Web, [http://solutions.journaldunet.com/0203/020322\\_xkms.shtml](http://solutions.journaldunet.com/0203/020322_xkms.shtml), 22 mars 2002, 10 mai 2002.

<sup>7</sup> Lindstrom, Pete, Special Report: The Language of XML Security, <http://www.networkmagazine.com/article/NMG20010620S0001>, Network Magazine 5 juin 2001

- a. Ces instances XML comprennent quatre types d'information <sup>8</sup>: le contenu chiffré du message lui-même, le contenu non chiffré pour des raisons de performance ou autre, de l'information décrivant la clé à utiliser et de l'information sur le destinataire du message chiffré;
- b. Le processus de chiffage XML suit les étapes de chiffage traditionnelles pour la cryptographie à clé publique. Tout d'abord on chiffre l'information en utilisant une clé secrète. Par la suite, la clé est chiffrée en utilisant la clé publique du destinataire. Le tout est transmis au destinataire; et
- c. Le destinataire suit la procédure inverse en utilisant sa clé privée pour déchiffrer la clé secrète. Avec la clé secrète, il pourra déchiffrer le message.

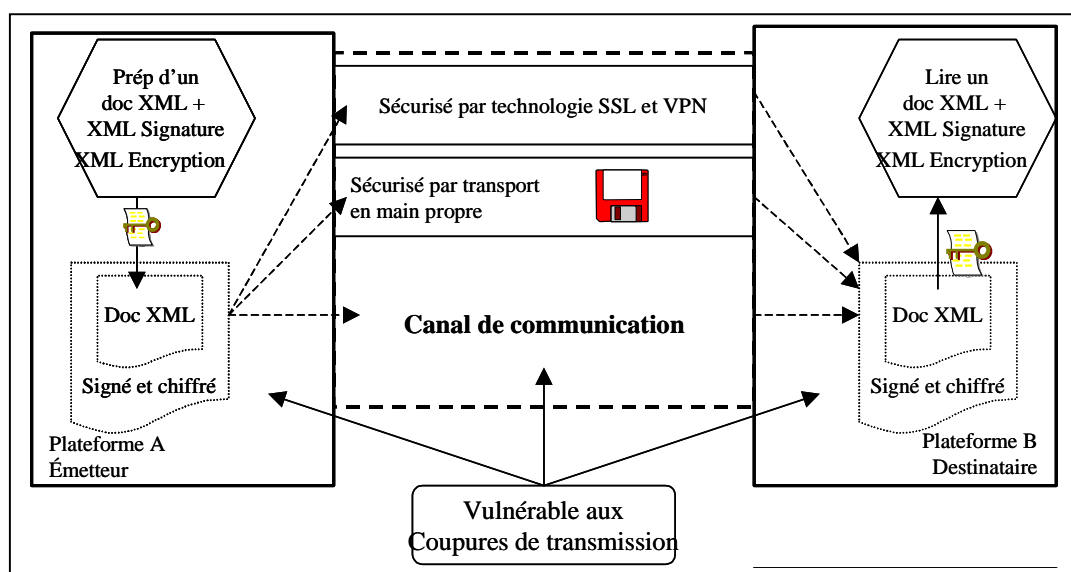


Figure 4-2 Document XML protégé sauf contre la vulnérabilité aux bris de transmission

### 4.3 XML pour accroître la sécurité

L'utilisation intelligente d'XML saura tirer avantage de ses caractéristiques pour accroître la sécurité. Par exemple, sachant qu'on est capable de sécuriser l'enveloppe d'un message avec les application XML Signature et Encryption, il est maintenant possible d'emprunter différents circuits de communication comme les courriers électroniques utilisant les protocoles « Simple Mail Transfert Protocol » (SMTP) et « Post Office Protocol » (POP). Cette confiance peut-être exploitée dans les technologies réseau. On se sert de l'identité de l'émetteur pour établir et lui assigner des droits d'accès<sup>9</sup>. Selon ses droits d'accès, l'utilisateur aura l'habilité de consulter les portails associés. L'habilitation, c'est l'assignation de droit d'accès spécifique à un usager qui

<sup>8</sup> Idem.

<sup>9</sup> Cover, Robin, Committee Specification Level Documents for the Security Assertion Markup Language (SMTP), < [http://xml.coverpages.org/ni\\_2002-04-20-a.html](http://xml.coverpages.org/ni_2002-04-20-a.html) > 5 mai 2002, 20 avril 2002.

aura été reconnu<sup>10</sup>. On peut déjà entrevoir le potentiel d'utilisation de ces outils dans un éventuel intranet de l'OTAN où tous les pays désirent contrôler l'échange d'information.

En acceptant la création d'un intranet pour l'OTAN, on pourra se servir d'un outil d'habilitation intitulé Security Assertion Markup Language (SAML). Le SAML est un dialecte XML pour exprimer des informations d'authentification sur des systèmes sécurisés. SAML permet de définir des droits utilisateurs génériques pour faciliter l'interopérabilité entre différentes applications. Cette technologie est à la base du Single Sign On (SSO). De sorte qu'à partir du moment où un usager est reconnu, son profil lui permettra automatiquement d'invoquer ses droits d'accès sur des portails spécifiques, sans avoir à lui-même y entrer son mot de passe<sup>11</sup>.

Les activations d'assertion définissent les propriétés et contraintes opérant sur les sujets, et définissent les droits d'exercice de ces sujets quant à l'accès aux ressources<sup>12</sup>. Ces relations sont démontrées dans la figure 4-3 On s'attend à ce que SAML devienne une norme industrielle endossée par l'Organization for the Advancement of Structured Information Standards (OASIS) vers juin 2002<sup>13</sup>.

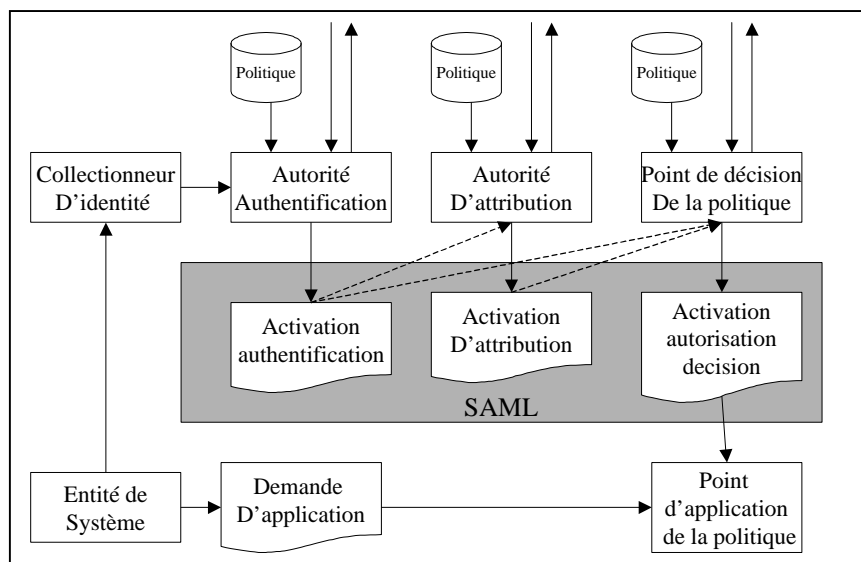


Figure 4-3 Les interactions SAML (Traduit et redessiné)<sup>14</sup>.

<sup>10</sup>OCTO Technology, Le Livre Blanc de la Sécurité, septembre 2001, page 17, ([www.octo.com/fr/techno/download/wp-secu.pdf](http://www.octo.com/fr/techno/download/wp-secu.pdf), 5 mai 2002).

<sup>11</sup> Idem, page 21.

<sup>12</sup> Agnoux, Hervé, Le consortium OASIS a déclaré que ses travaux autour de SAML étaient proches du niveau « maturité », <http://xmlfr.org/actualites/tech/020513-0001>, 13 mai 2002.

<sup>13</sup> Web and Web services Magazine, SAML Advances Single Sign-On Prospects, [http://www.fawcette.com/xmlmag/2002\\_03/magazine/departments/marketscan/SAML/](http://www.fawcette.com/xmlmag/2002_03/magazine/departments/marketscan/SAML/), 13 mai 2002.

<sup>14</sup> Idem.



### 4.3.1. Diverses Applications XML sécuritaires

D'autres applications connexes peuvent être utilisées pour exprimer et contrôler les droits d'accès à des données précieuses ou à des portails. Parmi ces applications XML, on y cite l'eXtensible rights Markup Language (XrML), l'Extensible Media Commerce Language (XCML) et l'Open Digital Rights Language (ODRL). Ce dernier permet de contrôler l'accès à un produit numérisé comme un enregistrement vidéo ou sonore<sup>15</sup>. On peut se représenter le principe de fonctionnement du ODRL en entourant l'objet précieux avec deux cercles. Le premier cercle spécifie les règles pour accéder et utiliser l'objet précieux. Le deuxième cercle étant le plus éloigné de l'objet, identifie les utilisateurs et les propriétaires de l'objet. Cette application est capable de donner l'accès à un client pour une période de temps pré établie ou même peut limiter le nombre de copies possibles de l'objet.

## 4.4 Sommaire sur la sécurité du XML

L'adoption de la technologie XML grandit sans cesse. Avec l'accroissement de l'expertise, des solutions palliatives diverses sont mises de l'avant pour rendre cette technologie plus robuste tout en maximisant sa caractéristique d'interopérabilité. Un système de sécurisation efficace insèrera graduellement la technologie XML selon la maturité des applications. On a dénombré des lacunes sécuritaires reliées au XML mais le XKMS avec les spécifications du XML Signature et XML Encryption est en voie de les éliminer. En déployant la technologie XML en tenant compte de ses caractéristiques, il est possible d'accroître la sécurité tout en fournissant des services non disponibles précédemment.

---

<sup>15</sup> Secure rights to digital assets with ODRL, <http://www.odrl.net/1.0/ODRL-10-HTML/ODRL-10.html>, 16 mai 02.

## Chapitre 5 – Les vocabulaires structurés du XML

### 5.0 Généralités

« La technologie du XML représente le format le plus versatile et le plus robuste pour l'échange d'information depuis le développement du standard Electronic Data interchange (EDI) », la firme Transentric<sup>1</sup> (traduction libre).

Cette remarque démontre le potentiel du XML et évoque son importance puisqu'on le compare à l'EDI qui est adopté par les grandes entreprises. Le succès de l'EDI est relié à la relation étroite qui existe avec l'Accredited Standards Committee (ASC) X12<sup>2</sup>. Par leurs efforts de normalisation, ils ont obtenu les consensus requis pour développer tout près de 275 transactions type depuis 1979. D'ailleurs c'est sensiblement l'approche que doit adopter tout secteur d'activité qui veut se munir de normes pour l'échange de données.

L'étude des facteurs communs rejoignant différent secteur d'activité permettra de saisir les caractéristiques des normes jugées avantageuses dans un vocabulaire structuré par le XML. Ces leçons apprises permettront de comprendre la situation à laquelle fait face les membres du MIP qui ont développé le MilML pour le secteur d'activité opérationnel militaire.

### 5.1 Identification des raisons motivant le développement des vocabulaires par couche de l'industrie.

C'est une vision utopique que d'espérer l'adoption universelle d'un langage numérisé commun à court et moyen terme. Présentement, il n'existe aucune norme globale capable de produire la gamme de transaction requise pour couvrir toutes les transactions d'affaires. Commensurablement, l'effervescence des milieux d'activités ne peut pas se permettre d'attendre plus longtemps l'émergence d'une norme globale. Il n'y a pas de solution miracle. La réalité est que chaque secteur d'activité doit se prendre en main et définir ses besoins d'échange de données par le biais du travail des comités existants comme le ChemXML<sup>3</sup> ou par la création de nouveaux groupes de travail diversifiés. Les intervenants recherchent des caractéristiques généralement semblables d'un vocabulaire à l'autre, en voici une liste :

- a. Un ensemble unifié de normes d'échange de données basées sur les pratiques courantes du secteur;
- b. Capable de réduire les coûts de production ou d'opération;

---

<sup>1</sup> Transentric, A White Paper, TranXML™ : The Common Vocabulary for Transportation Data Exchange, <<http://www.transentric.com/products/commerce/transWP.pdf>>, 5 mai 2002, page 1.

<sup>2</sup> Accredited Standard Committee, What is ASC X12, <<http://www.x12.org/x12org/about/whatis.html>>, 7 mai 2002.

<sup>3</sup> Chem eStandards™ Frequent Asked Questions, <[http://www.cidx.org/cidx/faq/faq\\_02.html](http://www.cidx.org/cidx/faq/faq_02.html)>, 6 mai 2002.

- c. Être capable de minimiser les erreurs manuelles lors de la capture de données;
- d. Capacité de rejoindre et communiquer de firme à firme, de firme à un marché, de marché à marché;
- e. En général, on tente d’accomplir la neutralité en ne favorisant pas de manufacturier en particulier;
- f. Permet l’indépendance d’action, sans avoir à utiliser les services d’une tierce partie dans une transaction;
- g. Particularité d’être lisible pour les humains et facilement traité par les SI; et
- h. Améliorer la rapidité et l’accès à l’information automatisée.

## 5.2 Les vocabulaires répertoriés

Qui sont ces intervenants? Ce sont les secteurs d’activités dans lesquels la communication et l’échange d’information tendent à être riches et dynamiques. Ce n’est malheureusement pas tous les secteurs d’activités qui ont la capacité de rassembler les ressources pour développer un vocabulaire satisfaisant leurs intérêts ! Parmi les milieux de travail d’envergure qui se sont engagés dans cette définition d’un vocabulaire électronique on y rencontre :

- a. ChemXML- l’industrie des produits chimiques;
- b. MathML- pour le milieu mathématique;
- c. TranML- pour le milieu des transports et de la chaîne logistique;
- d. HR-XML- pour le milieu des ressources humaines;
- e. ebXML- pour le milieu du commerce électronique;
- f. RosettaNet – pour le milieu des technologies de l’information, des composants électroniques et des semi-conducteurs;
- g. XBRL – pour le milieu financier des affaires (Business Reporting);
- h. LegalXML – pour le milieu de la justice et légal; et
- i. MilML- pour le milieu militaire qui sera discuté au prochain chapitre.

### 5.2.1 ChemXML

Les normes du Chem eStandards™ ont été développées par l'industrie et pour l'industrie des produits chimiques. Elles résultent d'un effort de coopération ouverte entre les producteurs de produits chimiques, le commerce électronique, les pourvoyeurs de services et du Chemical Industry Data eXchange (CIDX™)<sup>4</sup>. BASF, Dow et DuPont ont formé une alliance pour lancer de l'avant les normes XML de leur industrie qui ont été rapidement adoptées durant les derniers mois de l'an 2000<sup>5</sup>. Par exemple, la firme Air Product, pressés par Intel<sup>6</sup>, a procédé à la conversion de ses échanges électroniques pour maintenir sa part de marché. Le gain par Intel est une augmentation de données devenues plus simples et souples. Ainsi les résultats d'un échantillon sont utilisés pour émettre un certificat d'analyse de la pureté des gaz en format CIDX envoyé par le File Transfert Protocol (FTP) à BizTalk. BizTalk convertit le document en format XML puis envoi une copie par l'internet. Voici le genre d'information spécifiquement traité par le Chem eStandards™<sup>7</sup>:

- a. Emballage et la sécurité des produits dangereux;
- b. Bon de commande et catalogue; et
- c. Logistique et manipulation des produits dangereux.

### 5.2.2 MathML

MathML est probablement le vocabulaire le plus avancé et le mieux reconnu du groupe W3C. On s'en sert déjà depuis quelques années et un document exhaustif est disponible au travers l'internet<sup>8</sup>. L'objectif principal du MathML est de rendre accessible les mathématiques par l'internet de la même façon que les textes le sont avec HTML.

Le MathML permet de transmettre de l'arithmétique, de l'algèbre, de la logique et les relations, du calcul différentiel et intégral, les vecteurs, les théories, les séquences et les séries, les fonctions élémentaires classiques, les statistiques et l'algèbre linéaire. Pour se servir du MathML, on doit suivre une notation qui inclut des règles de grammaires et de syntaxes définies<sup>9</sup>. Il y a environ 30 balises MathML qui décrivent les structures de

---

<sup>4</sup> Le Chemical Industry Data eXchange (CIDX™), Un livre blanc sur le CIDX, <[http://www.cidx.org/cidx/res\\_library/index.html](http://www.cidx.org/cidx/res_library/index.html)>, 6 mai 2002.

<sup>5</sup> How CHEM eStandards™ Were Developed, <[http://www.cidx.org/cidx/commerce/how\\_dev.html](http://www.cidx.org/cidx/commerce/how_dev.html)>, 6 mai 02.

<sup>6</sup> Koller, Mike, Pushed Off The Fence, <<http://www.internetweek.com/newslead01/lead112601.htm>>, 26 November 2001

<sup>7</sup> Chem eStandards™ Frequently Asked Questions, <[http://www.cidx.org/cidx/faq/faq\\_02.html](http://www.cidx.org/cidx/faq/faq_02.html)>, 6 mai 2002.

<sup>8</sup> W3C Recommendation 21 Feb 01, <[www.w3.org/TR/2001/REC-MathML2-20010221](http://www.w3.org/TR/2001/REC-MathML2-20010221)>, 6 mai 2002.

<sup>9</sup> The Goals of MathML, <<http://www.w3.org/Math/mathml-faq.html#goals>>, 6 mai 2002.

notation abstraite tandis qu'environ 150 autres constituent une façon d'exprimer le contenu d'une expression sans ambiguïté.

### 5.2.3 TranXML

TranXML est un vocabulaire commun d'échange de données pour les transports développé pour soutenir la chaîne d'approvisionnement logistique. La particularité réclamée est d'être un secteur d'activité qui accède de façon verticale aux autres secteurs d'activités. La solution TranXML vante ses mérites comme un complément à l'EDI. Elle inclut le transport ferroviaire, par voie de surface, les inventaires de stock, le poids des marchandises et beaucoup d'autres<sup>10</sup>.

Avec raison, le domaine des transports touche à plusieurs secteurs et doit aisément rejoindre ses clients. On ne sera pas surpris que les autres secteurs d'activités ait incorporé leur processus d'approvisionnement dans leurs vocabulaires. L'influence de ses pratiques sur la sémantique électronique est perçue comme un risque pour le domaine du transport qui craint qu'on « Demande aux transporteurs d'adopter et de supporter des normes industrielles inadéquates pour leurs besoins »<sup>11</sup>. En riposte, Transentric se dit un porte-parole pour la défense de l'industrie du transport et parraine le développement du TranXML<sup>TM</sup><sup>12</sup>.

### 5.2.4 HR-XML

HR-XML est un consortium indépendant, à but non-lucratif qui soutien les développement de normes XML pour l'automatisation des procédés reliés aux activités des ressources humaines<sup>13</sup>.

Les bienfaits de ces normes s'appliqueront dans les échanges électroniques pour la dotation en ressources humaines, les échanges impliquant le personnel avec des qualifications uniques, les cartes de temps des employés, leurs salaires et bénéfices sociaux<sup>14</sup>. Des normes plus détaillées du HR-XML sont élaborées dans le Staffing Industry Data Exchanges Standards (SIDES)<sup>15</sup>. Ce service visera les agences gouvernementales et les nombreux pourvoyeurs de service de dotation

---

<sup>10</sup> Transentric, A White Paper, TranXML<sup>TM</sup> : The Common Vocabulary for Transportation Data Exchange, <<http://www.transentric.com/products/commerce/transWP.pdf>>, 5 mai 2002, page 3.

<sup>11</sup> Idem, page 1.

<sup>12</sup> Idem, page 1.

<sup>13</sup> HR-XML Consortium, About HR-XML, <<http://www.hr-xml.org/channels/about.cfm>>, 5 mai 2002.

<sup>14</sup> HR-XML Consortium, SIDES Fact Sheet, <<http://www.hr-xml.org/sides/sidesfactsheet.cfm>>, 5 mai 2002.

<sup>15</sup> HR-XML Consortium, Guide to SIDES, <<http://www.hr-xml.org/sides/GuideToSIDES.pdf>>, 5 mai 2002, page 2.

### 5.2.5 ebXML

ebXML est l'acronyme utilisé pour désigner Electronic Business using eXtensible Markup Language. Ce vocabulaire est une série de spécifications parrainé par United Nations Centre For Trade Facilitation and Electronic Business (UN/CEFACT) et OASIS qui permet aux entreprises de toute grandeur de se servir de l'Internet pour la réalisation de leur commerce malgré leur situation géographique.

Avec ebXML, les entreprises auront accès à « une méthode normalisée pour échanger des messages d'affaires, poursuivre leurs échanges avec leurs partenaires, transmettre des données selon une terminologie commune ainsi que de définir et d'enregistrer leurs procédures d'affaires »<sup>16</sup>. Plusieurs comités interagissent entre eux pour développer un processus commun d'affaires tel qu'exploité par ebXML. On y retrouve en autres EDI, BOLERO.NET et X12<sup>17</sup>.

### 5.2.6 RosettaNet

« RosettaNet est un processus normalisé pour l'échange d'information d'affaires qui relie toutes les entreprises impliquées dans l'approvisionnement et l'achat des nouvelles technologies »<sup>18</sup>. C'est un consortium d'intervenants à but non lucratif comprenant tout près de 400 entreprises à la fine pointe des technologies de l'information, des composantes électroniques, des manufacturiers de semi-conducteurs et des fournisseurs de solutions (Solution Provider SP).

En utilisant le processus électronique normalisé RosettaNet, il est possible d'améliorer les relations d'affaires, d'économiser les coûts et d'augmenter la productivité. RosettaNet cherche à satisfaire les besoins des manufacturiers de la chaîne d'approvisionnement, des distributeurs, des revendeurs et des utilisateurs<sup>19</sup>.

---

<sup>16</sup> ebXML, Enabling A Global Electronic Market – General Information  
<<http://www.ebxml.org/geninfo.htm>>, 6 mai 2002.

<sup>17</sup> ebXML Catalog of Common Business Processes v1.0, 11 mai 2001,  
<<http://www.ebxml.org/specs/bpPROC.pdf>>, page 9.

<sup>18</sup> RosettaNet Overview,  
<<http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?Container=com.webridge.entity.Entity%5B0ID%5B36249C27FC2BD411841F00C04F689339%5D%5D>>, 6 mai 2002.

<sup>19</sup> RosettaNet, Standards,  
<[http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?expanded=com.webridge.entity.Entity\[OID\[5F6606C8AD2BD411841F00C04F689339\]\]&container=com.webridge.entity.Entity\[OID\[5F6606C8AD2BD411841F00C04F689339\]](http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?expanded=com.webridge.entity.Entity[OID[5F6606C8AD2BD411841F00C04F689339]]&container=com.webridge.entity.Entity[OID[5F6606C8AD2BD411841F00C04F689339])>, 5 mai 02

### 5.2.7 XBRL

L'acronyme XBRL vient de la terminologie Extensible Business Reporting Language. XBRL est un vocabulaire de type XML développé pour le monde financier qui inclut entre autre les plans d'affaires des entreprises, les états financiers pour des entreprises privées et publiques<sup>20</sup>. Les grandes firmes d'expert conseil opérant dans la chaîne d'approvisionnement bénéficieront de ce vocabulaire comme un outil potentiellement plus accessible que le logiciel EDGAR (US) utilisé dans les rapports annuels des compagnies publiques.

### 5.2.8 LegalXML

LegalXML, comme son nom le suggère, est un effort de normalisation d'un vocabulaire XML pour l'échange de données légales<sup>21</sup>. Ce groupe formé d'experts légaux et techniques, est aussi une branche participante au sein d'OASIS, un organisme à but non-lucratif qui parraine le développement, la convergence et l'adoption des normes d'affaires électronique. LegalXML est composé d'avocats, des développeurs, des vendeurs d'application, des agences gouvernementales et des membres académiques. Les objectifs de LegalXML est la production de normes servant en autres à l'archivage électronique de la cour, les documents de cour, les citations légales, les transcriptions, le système de renseignement sur la justice criminelle.

## 5.3 Sommaire

Le nombre de vocabulaire XML ne cesse de croître et s'étend à de nombreux secteurs d'activités. Il faut investir des ressources car les travaux de normalisation sont longs et pénibles car ils exigent d'innombrables consultations. Les attentes doivent être mesurées car malgré les résultats positifs on invite à la prudence. Les vérificateurs gouvernementaux américains ont édicté leur position sur le statut du XML en avril 2002 dans les rapports du General Accounting Office (GAO) et du National Automated Clearing House Association (NACHA)<sup>22</sup>. La situation des vocabulaires LegalXML, le HR-XML et le XBRL n'était pas aussi stable qu'on veut l'afficher. Car malgré leur bon concept, l'aspect exécution est pauvre. Pour d'autres secteurs d'activités la compétition existe pour faire approuver des normes qui ne sont pas pas aussi représentatives que présentées. On cite entre autre une compétition impliquant ebXML, RosettaNet, ainsi que le SOAP, WSDL et UDDI.

Malgré tout, on reconnaît le bien fondé de poursuivre les efforts de normalisation. C'est sur cette voie que les forces militaires doivent continuer à améliorer et à perfectionner leur vocabulaire, le MilML.

---

<sup>20</sup> XBRL, Overview Fact Sheet, < <http://www.xbrl.org/Overview.htm> > , 5 mai 2002.

<sup>21</sup> OASIS, About LegalXML, < <http://www.legalxml.org/about/index.shtml> > , 6 mai 2002.

<sup>22</sup> Kotok, Alan, Government and Finance Industry Urge Caution on XML, April 24, 2002. < [www.xml.com/pub/a/2002/04/24/gaonacha.html](http://www.xml.com/pub/a/2002/04/24/gaonacha.html) > , 6 mai 2002.

## **Chapitre 6 – MilML – Un vocabulaire structuré pour l’environnement militaire**

### **6.0 Généralités**

La numérisation du champ de bataille pour les opérations militaires devient rapidement une réalité pour les forces alliées opérant au sein de l’OTAN. Une multitude de SI sont ainsi créés pour traiter une quantité sans cesse grandissante de données entreposées dans différentes bases de données. Malheureusement ces données n’ont du sens que dans le contexte du SI dans lequel elles sont produites. Pour les comparer et les traduire dans un contexte plus général, il faut les définir dans une structure commune rendant possible l’interopérabilité. La structure du MilML est certainement une solution qui accélèrera l’automatisation de l’échange d’information au niveau international applicable aussi aux SI nationaux. Il appartient à l’environnement militaire de définir les spécifications propres à son vocabulaire.

### **6.1 Modèle d’échange de données commun – LC2IEDM/LFDM**

Le MilML, comme les autres vocabulaires XML discutés au chapitre 5, doit obtenir le consensus des membres du MIP pour récolter les bénéfices liés à l’interopérabilité. Une stratégie pour développer un vocabulaire structuré repose sur l’acceptation commune d’un modèle représentatif du secteur d’activité par exemple l’environnement militaire. Avec ce modèle représentatif du processus des opérations militaires, il devient possible d’établir des règles d’affaires qui seront inscrites dans le vocabulaire structuré MilML. Adroitement, malgré les différentes cultures et traditions militaires, les membres du MIP ont cerné une structure globale qui décrit adéquatement ces activités militaires. Ce modèle commun est le Land C2 Information Exchange Data Model (LC2IEDM) qui devient la base de l’interopérabilité pour les SI des pays de l’OTAN. C’est ce modèle qui servira de base à la structuration des règles d’affaires du MilML.

Le modèle LC2IEDM peut être étendu pour inclure des caractéristiques nationales par l’addition de champs unique et discrétionnaire. Ainsi, on a formé un modèle des FT, le « Land Force Data Model (LFDM) ». La figure 6-1 représente la relation de réciprocité entre les deux modèles qui permet au besoin, de cerner la nature d’une donnée selon son appartenance à un de ces modèles. Chaque donnée d’OPERA appartient à un ou deux de ces modèles. Au sein des FC, il y a d’autres modèles de données canadiens, toutefois leur structure n’a pas d’incidence sur ce projet d’interopérabilité. Ils sont le « Air Environment Data Model (AEDM) » ainsi que le « Maintenance Support Data Model (MSDM) », qui ne seront plus discutés.



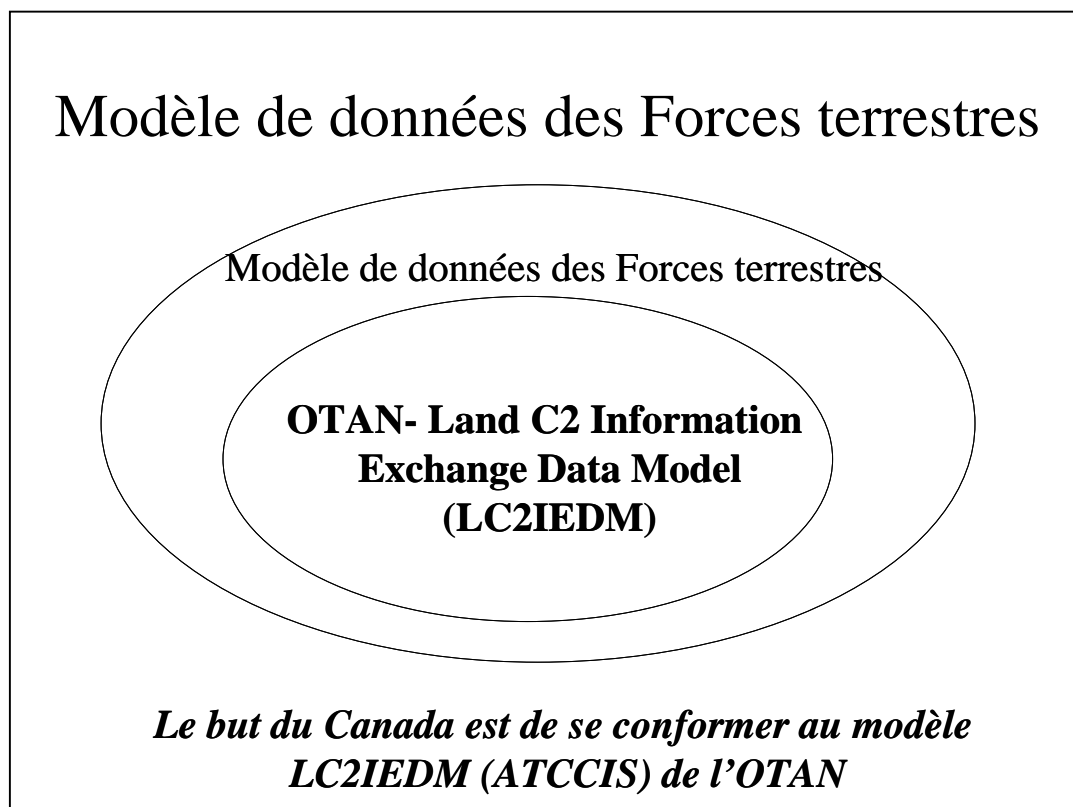


Figure 6-1 Modèle de données des forces terrestres (traduit)<sup>1</sup>

## 6.2 MilML

### 6.2.1 Le MilML construit pour les opérations tactiques

Le vocabulaire structuré MilML cherche surtout à définir l'environnement tactique des opérations militaires pour réaliser l'interopérabilité multinationale. Cette précision est importante car à l'inverse, l'information générée par les SI de garnison correspond à une structure de données différente qui n'est pas encore visé par le MilML. Cette précedence déterminera la structure de l'information MilML, la portée de l'information traitée, la pertinence et le niveau de détails souhaités pour les opérations tactiques. On doit être aisément capable de saisir le sens de l'information et l'exécuté rapidement. La première version du MilML<sup>2</sup> a justement livré un document basé sur les données tactiques donc la branche souche est l'élément « opération » ou « Ops » qui s'exprimerait ainsi:

<sup>1</sup> Weisman, Robert, Presentation on AIME – Environment Phase 2 Proof of Concept, Groupe CGI Inc, 16 March 2000.

<sup>2</sup> CGI, Military Mark up Language - A proposal for Military Interoperability Language based on XML and the NATO LC2IEM, draft version 1.7, 31 mai 2000, page 9.

<Ops>

... tous les éléments MilML sont insérés entre ces balises

</Ops>

## 6.2.2 Les spécifications techniques du MilML

Lors de la création du vocabulaire MilML on s’est servi de la version « XML Schema 1.0 »<sup>3</sup>, XML Specification 1.0 ainsi que des recommandations du « Namespaces » adoptées par le W3C en date du 14 janvier 1999<sup>4</sup>. Le préfixe « Namespace » des FT est *land.dnd.ca*. De même, l’emploi de plusieurs acronymes est aussi propre au MilML, pour lequel il est devenu valable de créer l’annexe A établissant la convention entre les noms logiques et les noms physiques<sup>5</sup>. Par exemple « operation » et « operational » deviennent « op » et « oper ». Ce document capture un ensemble impressionnant de termes utilisés pour l’interopérabilité militaire.

## 6.2.3 La structure physique du MilML- entités

Puisque la structure du LC2IEDM est la source de référence du MilML, on considère que son contenu est sensiblement le même. Le LC2IEDM est fondé sur des classes d’objet abstrait. Dans le modèle de donnée du MilML on délimite ainsi dix entités indépendantes. Ces entités fondamentales sont LOCATION, CAPACITÉ, ACTION, OBJET-ITEM, OBJET-TYPE, LISTE, CONTEXTE, RÉFÉRENCE, DONNÉE-RAPPORT et RÈGLES D’ENGAGEMENT dont la description des cinq premières a été inclut à l’annexe B. On reconnaîtra l’importance des entités location, capacité et actions pour décrire les manœuvres militaires plus applicables aux SI ATS et SAS. Tandis que dans l’interaction des SI OPERA et CMS, l’importance est accordée à la notion relationnelle d’OBJECT-TYPE (OT) et OBJECT-ITEM(OI).

## 6.2.4 Les OBJETS TYPE et les OBJETS ITEM du MilML

La distinction entre les entités OBJET-ITEM et les OBJET-TYPE est vitale dans la compréhension des enjeux du chapitre 7. Implicitement, les données reliées aux OT seront statiques. Par exemple un OT serait ce qui définit généralement un bataillon d’infanterie. De l’autre côté, il y a la donnée OI plus dynamique qui spécifie un bataillon selon son identité comme le 1<sup>er</sup>R22R ou le 2 RCR. La force de cette relation réside dans la capacité d’exprimer le même OI soit l’unité 1<sup>er</sup> R22R ou le 2 RCR se conformant aux caractéristiques OT des unités d’infanteries. On dira que les OI hériteront des attributs des OI.

---

<sup>3</sup> Idem, page 8.

<sup>4</sup> Extrait de “Namespaces in XML”, <http://www.w3.org/TR/1999/REC-xml-names-19990114>, Copyright © 1999 W3C (MIT, INRIA, Keio), W3C, W3C RECOMMENDATION, 6 mai 2002.

<sup>5</sup> CGI, Military Mark up Language, Annexe A.

Toutes les relations entre les données militaires ont ainsi été définies. La figure 6-2 représente graphiquement la relation entre les entités clés OT et les OI en plus d’élaborer sur la hiérarchie existante avec leur sous entités sous jacente. Les sous-entité OT et OI sont définies dans les tableaux B-2 et B-3 de l’annexe B. Pour bien visualiser ce modèle, il faut partir du haut puis définir par couche l’objet qu’on désire représenter. Par exemple les organisations peuvent être des brigades, des unités ou des compagnies. L’unité 1<sup>er</sup>R22R est l’ORGANISATION sous l’entité OI. En même temps, la structure du 1<sup>er</sup>R22R se conforme aux unités d’infanterie type détaillé sous l’ORGANISATION-TYPE du côté OT.

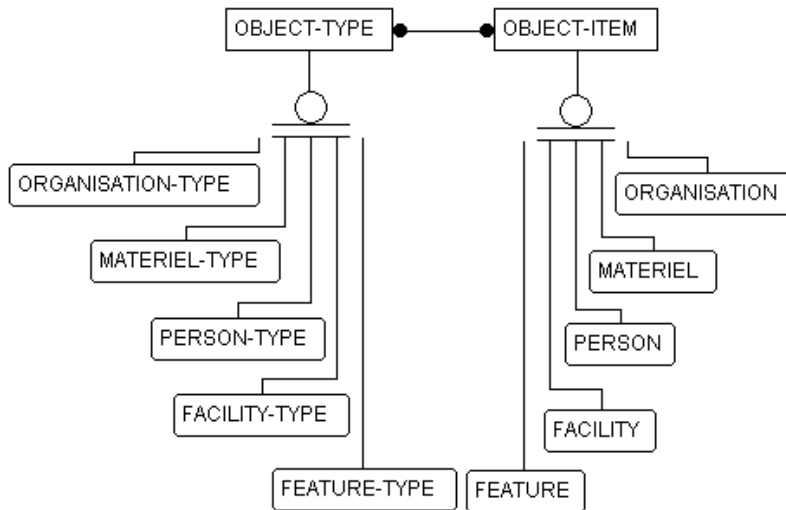


Figure 6-2 Hiérarchie des objets typiques du « LC2IEDM »<sup>6</sup>

En poussant cet exemple au niveau plus bas, on représentera l’unité 2 RCR avec ses caractéristiques OT et OI selon chacune de ses compagnies et de ses pelotons. On peut ajouter de l’information dans les branches MATERIEL et PERSON. Chacune de ses branches pourra à leur tour être redéfinies. Par exemple, à la figure 6-3 on a inséré un TTB de type Equipment Configuration Code (ECC) 654321 et de Canadian Forces Registration CFR 67-12345 appartenant à l’unité 2 RCR. Dans cet exemple, toutes ces données sont reliées sauf les sous entités OT et OI (PERSON, FACILITY et FEATURE) pour lesquels aucune valeur n’a été assignée.

<sup>6</sup> Idem, page 15.

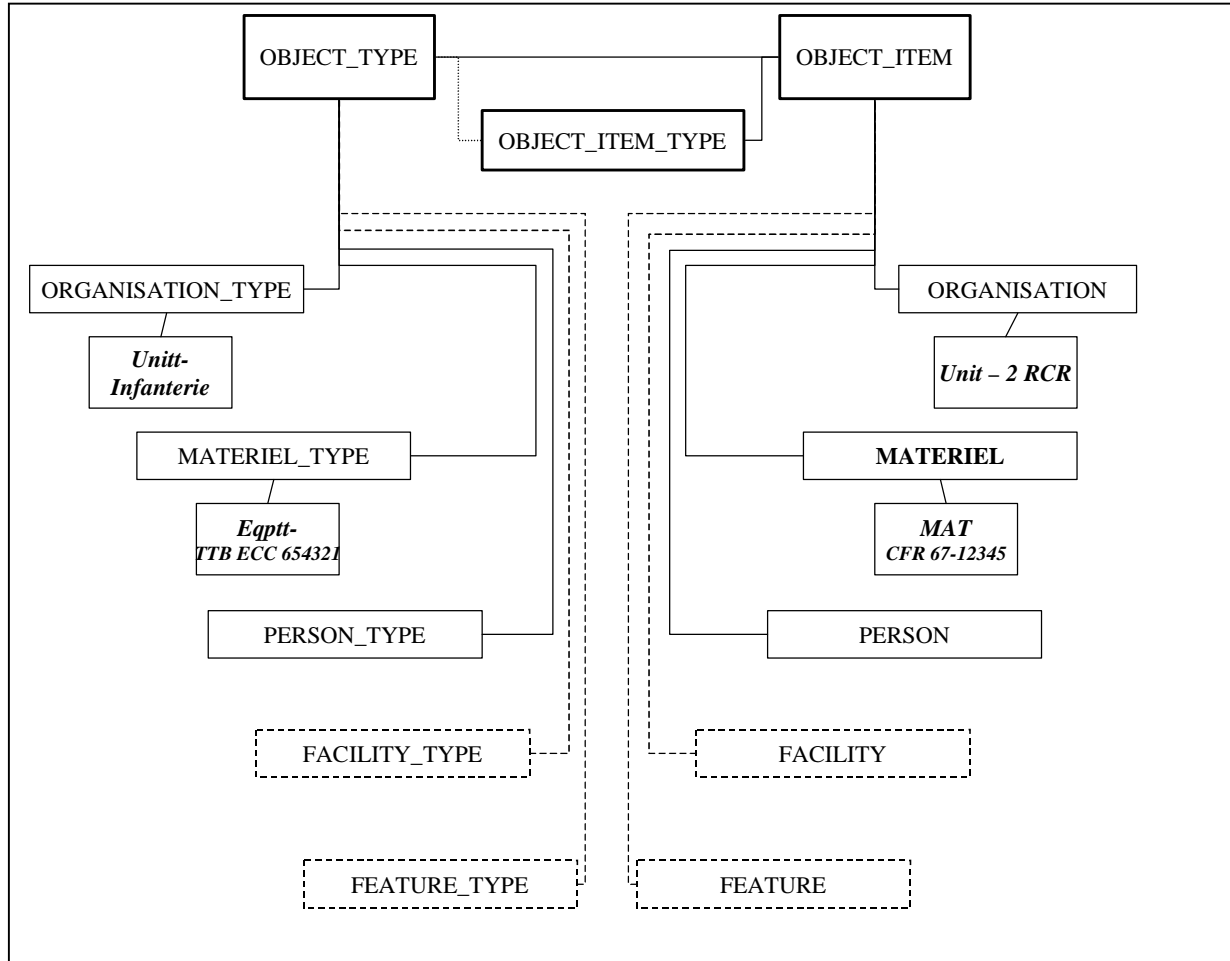


Figure 6-3 Exemple de l'interrelation de la structure MilML

### 6.3 Sommaire sur le MilML

Ce chapitre s'est efforcé de démontrer l'énorme travail déjà entrepris dans la normalisation d'un vocabulaire MilML. Les concepts sont pleinement élaborés dans les groupes de travail de l'OTAN. Des exercices d'interopérabilité ont lieu régulièrement. Le document pris en référence sur le MilML est une ébauche d'un travail fait pour les FT. Il y a beaucoup d'autres détails qui vont en profondeur dans les branches des entités clés. Il faut mentionner que les entités OI et OT ne sont qu'une facette du MilML, et que pour les opérations d'autres entités clé sont déjà définies. Ce document deviendra une source de référence essentielle pour tous les nouveaux SI qui seront implémentés dans les FT. Le langage MilML crée des conditions favorables au développement et à la poursuite de l'interopérabilité. Théoriquement simple, un aspect qui n'en demeure pas moins difficile sera de démontrer la capacité du XML comme outil d'interopérabilité avec les systèmes se servant d'ancienne technologie.

## Chapitre 7 – Une solution appliquée pour les SI entre OPERA et CMS

### 7.0 Généralités

Une définition de l'interopérabilité offerte par le « NATO Interoperability Planning Document (NIPD) » capture très bien l'essence du climat de coopération qui doit régner si on espère développer une synergie entre tous les systèmes d'information.

« C'est l'aisance des systèmes, unités ou forces à fournir et à accepter des services assortis à d'autres systèmes, unités ou forces en utilisant ces services échangés pour leur permettre d'opérer effectivement ensemble »<sup>1</sup> (traduction libre).

La documentation du MIP et du MilML guide les efforts du Canada de normalisation du partage d'information au sein des niveaux de commandement des FT. Une manière d'arriver à l'interopérabilité serait de lier entre eux tous les SI des FT et des FC afin qu'ils puissent partager leurs données, selon un modèle semblable à la figure 7-1:

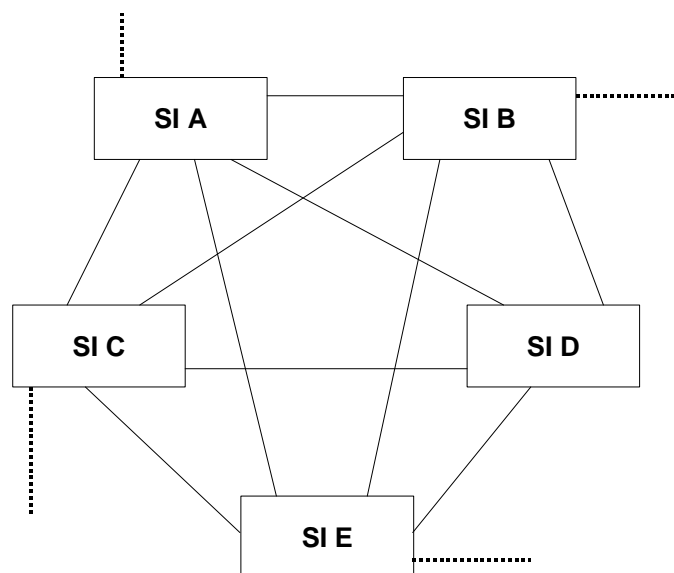


Figure 7-1 Modèle Multi-Systèmes Interconnectés (Redessiné et adapté)<sup>2</sup>

Le désavantage de cette architecture est la lourdeur de ces connexions pour interconnecter ces systèmes, de l'ordre de  $n(n-1)/2$  où  $n$  est le nombre de systèmes.

À l'opposé le modèle de la figure 7-2, favorise plutôt une interconnexion des systèmes qui partage une base de données centrale qui contient toutes les informations. En raison du nombre de connexions moins nombreuses, soit  $n$  connexions, le système est beaucoup plus simple à gérer.

<sup>1</sup> Présentation de DIRT au LF Command Council, L'Interopérabilité une Réalité, 28 janvier 2000.

<sup>2</sup> Groupe CGI, "Generic Enterprise Data Integration System(GEDIS) General Architecture V1.2", 26 mars 2002.

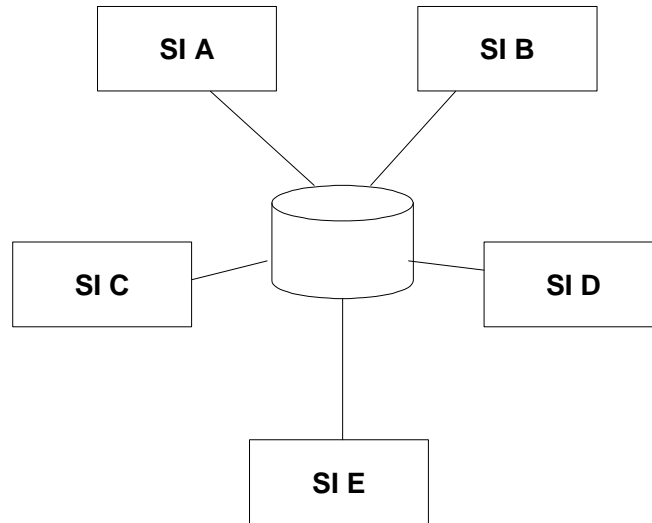


Figure 7-2 Modèle Multi-systèmes concentré sur une base de données (Redessiné et adapté)<sup>3</sup>

### 7.0.1 Operational Data Store (ODS)

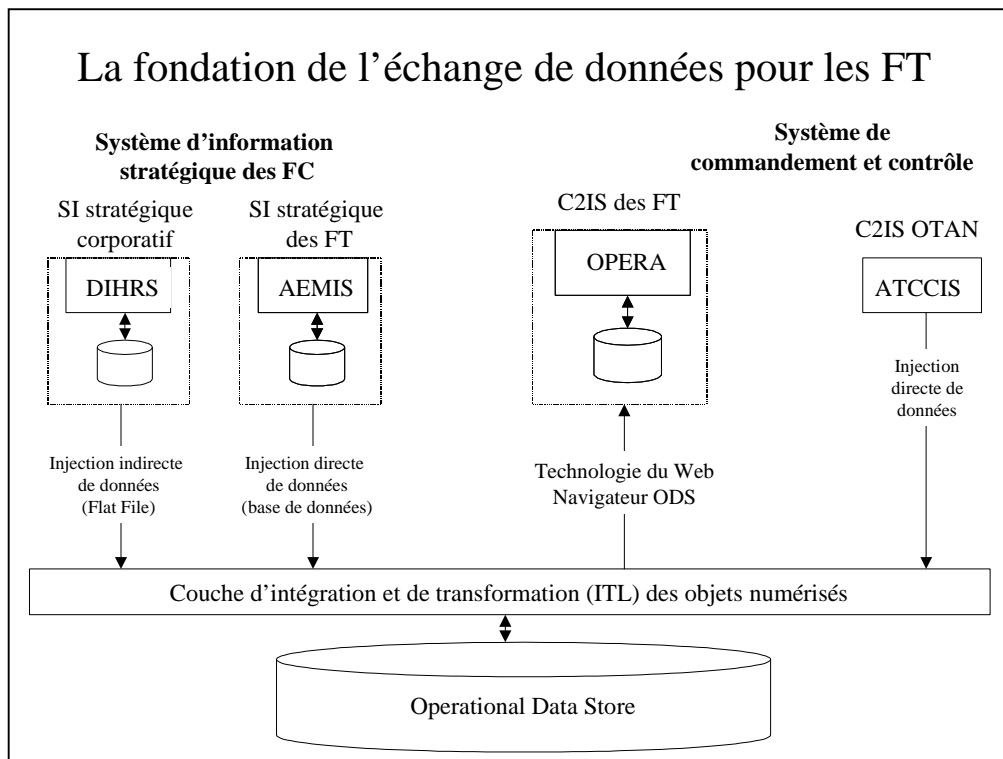


Figure 7-3 ODS- La fondation des échanges de données pour les FT<sup>4</sup>

<sup>3</sup> Groupe CGI, Generic Enterprise Data Integration System(GEDIS) General Architecture V1.2, 26 mars 2002

<sup>4</sup> Weisman, Robert, "AIME Environment Phase 2 – Proof of Concept Brief", 16 mars 2000.

C'est ce concept d'une base de donnée centrale appelée Operational Data Store (ODS) qui a été implémenté par le « Army Integrated Management Environment (AIME) »<sup>5</sup> donc l'objectif est d'accomplir le défi d'intégration de tous les SI au service de l'armée. La représentation graphique à la figure 7-3 démontre l'importance de l'ODS pour les FT. Cette base de données sert de répertoire principal pour emmagasiner une foule d'information dont l'établissement des organisations et leur données opérationnelles incluant les SI corporatifs des FC. L'ODS devient la fondation de l'interopérabilité au service de tous les SI des FT capables de s'y brancher. Il sera noté qu'OPERA est physiquement reliée à l'ODS alors que CMS ne l'est pas.

L'hypothèse de l'expérimentation qui va suivre sera de démontrer la pertinence d'utiliser l'outil XML pour créer une interface reliant le système OPERA de technologie récente et l'application CMS de technologie plus ancienne. Voici la démarche qui sera employée :

- a. Description du problème d'interopérabilité entre OPERA et CMS.
- b. Description du système d'information OPERA
- c. Description du système d'information CMS.
- d. Détail de la démarche d'expérimentation et de modélisation entre les deux systèmes d'information.
- e. Expérimentation – démonstrateur technologique.
- f. Recommandations XML liées à l'interopérabilité entre Opéra et CMS.

---

<sup>5</sup> DLIR, Army Integrated Management Environment (AIME) Overall Technical Architecture v 1.7, 12 March 2001, page 40.

## 7.1 Description du problème d'interopérabilité entre OPERA et CMS

La partie expérimentale de ce projet porte sur deux systèmes d'information de la force terrestre incompatibles au niveau de l'échange d'information. Il s'agit de l'Operational Planning Environment and Reference Application (OPERA) et du Communication Management System (CMS).

L'application OPERA détient des informations dans sa base de données sous-jacente qui sont utiles pour l'application CMS. Il s'agit des données sur l'ordre de bataille (ORBAT) qui contient l'information sur les unités et des données sur les véhicules utilisés par ces unités.

Les deux SI résident sur des systèmes d'exploitation différents, soit Windows NT 4 pour OPERA et SCO UNIX 5.0.2 pour CMS. Les deux SI sont des applications distribuées.

OPERA est exécuté dans un contexte réseau à l'intérieur d'un quartier général alors que CMS n'a pas de connexion sur le monde extérieur. Les deux systèmes ne peuvent donc pas se rejoindre par protocole de communication dans leur configuration actuelle.

Voici une modélisation simplifiée de notre problème:

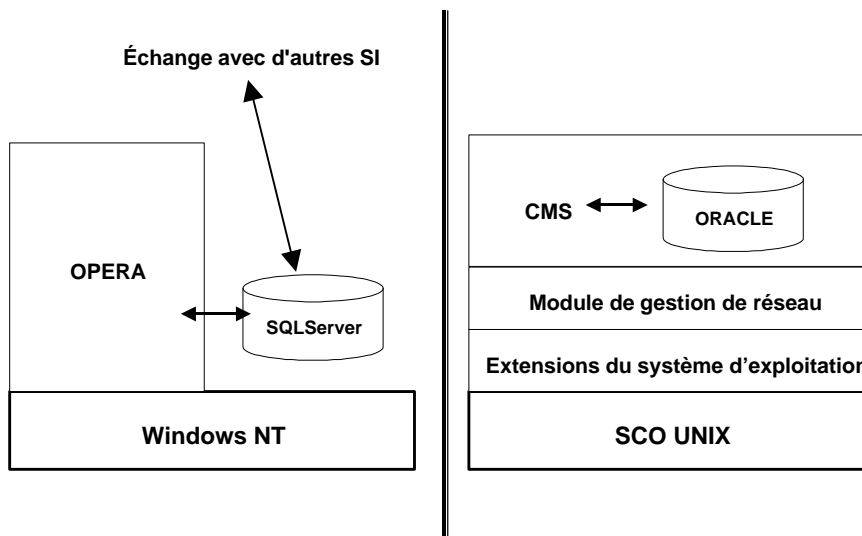


Figure 7-4 Problématique d'interopérabilité entre OPERA et CMS

Alors que la base de données SQL Server qui supporte OPERA peut établir des connexions en tant que serveur vers d'autres applications clientes, soit localement ou par réseau, le système distribué de IRIS ne permet à la base de données ORACLE qui l'accompagne que de communiquer avec son application mère CMS. Le module de gestion de réseau de concert avec les extensions du système d'exploitation empêche l'ouverture à d'autres connexions.



Le modèle des deux bases de données au niveau des tables et des champs n'est pas correspondant. C'est à dire que le nom donné à un item dans une base de données est différent de celui attribué dans l'autre, ajoutant une difficulté supplémentaire pour assurer un transfert d'information fiable.

Le problème consiste donc à trouver une façon d'extraire l'information pertinente de la base de données supportant OPERA et de l'injecter dans la base de données supportant CMS tout en effectuant une transformation de structure afin de respecter la différence de modèle entre les deux.

## 7.2 Description du système d'information OPERA



L'Operational Planning Environment and Reference Application (OPERA) est un outil de gestion opérationnelle au niveau du Quartier général. Ce SI est conçu pour aider à la planification opérationnelle dans le cadre d'opérations militaires. Il est toujours en développement au moment d'écrire ces lignes.

OPERA est une application écrite en langage Delphi, s'exécutant sur un système d'exploitation Windows NT. Il tire ses données d'une base de données SQLServer selon le modèle de données des Forces terrestres, qui lui est conforme au "Land Command and Control and information Exchange Data Model" de l'OTAN, ce qui lui assure une interopérabilité interne au sein de la Force terrestre et externe au sein des Forces alliées.

Il utilise la technologie DCOM et XML afin d'établir des connexions et d'effectuer des échanges de données avec d'autres applications.

Cet effort d'établir une application qui s'intègre dans un contexte d'interopérabilité soutient le contexte d'un environnement électronique intégré dont voici un modèle:

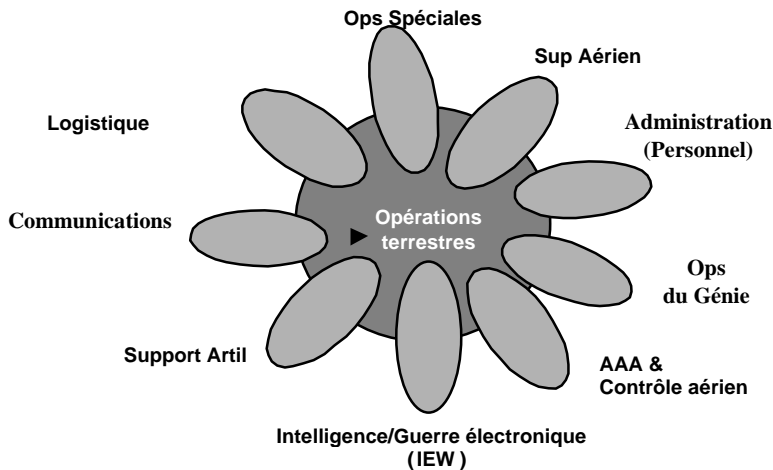


Figure 7-5 L'environnement électronique intégré <sup>6</sup>

<sup>6</sup> Groupe CGI, "Introduction to ATCCIS Information Session presented to Land Forces Staff", Mars 2000

### 7.2.1 Les composantes d' OPERA

L'application OPERA réunit un ensemble d'information et d'outils afin de permettre leur utilisation au sein du Système d'information de Commandement et Contrôle et de la Force terrestre (LFC2IS). Elle contient les composantes suivantes<sup>7</sup>:

- a. Base de données Opérationnelle ou "Operational Data Base" (ODB)
  - (1) Systèmes d'information National
    - (a) AEMIS
    - (b) HRMS (PeopleSoft)
    - (c) MASIS
    - (d) PlannExpert
    - (e) CFSSU
- b. Coffret Électronique (EBB)
  - (1) Navigateur d'équipement
  - (2) Navigateur d'ORBAT
  - (3) Navigateur de tâches
  - (4) Outils d'état-major
- c. Planificateur logistique
- d. Planificateur de mouvement routier
- e. Planificateur de mouvement aérien
- f. Planificateur de mouvement ferroviaire

---

<sup>7</sup> DCIT, "DLCI Command Support Brief to 2 CMBG", 6 septembre 2001.

## 7.2.2 L'interface utilisateur

Voici un aperçu de l'interface utilisateur (GUI) de OPERA

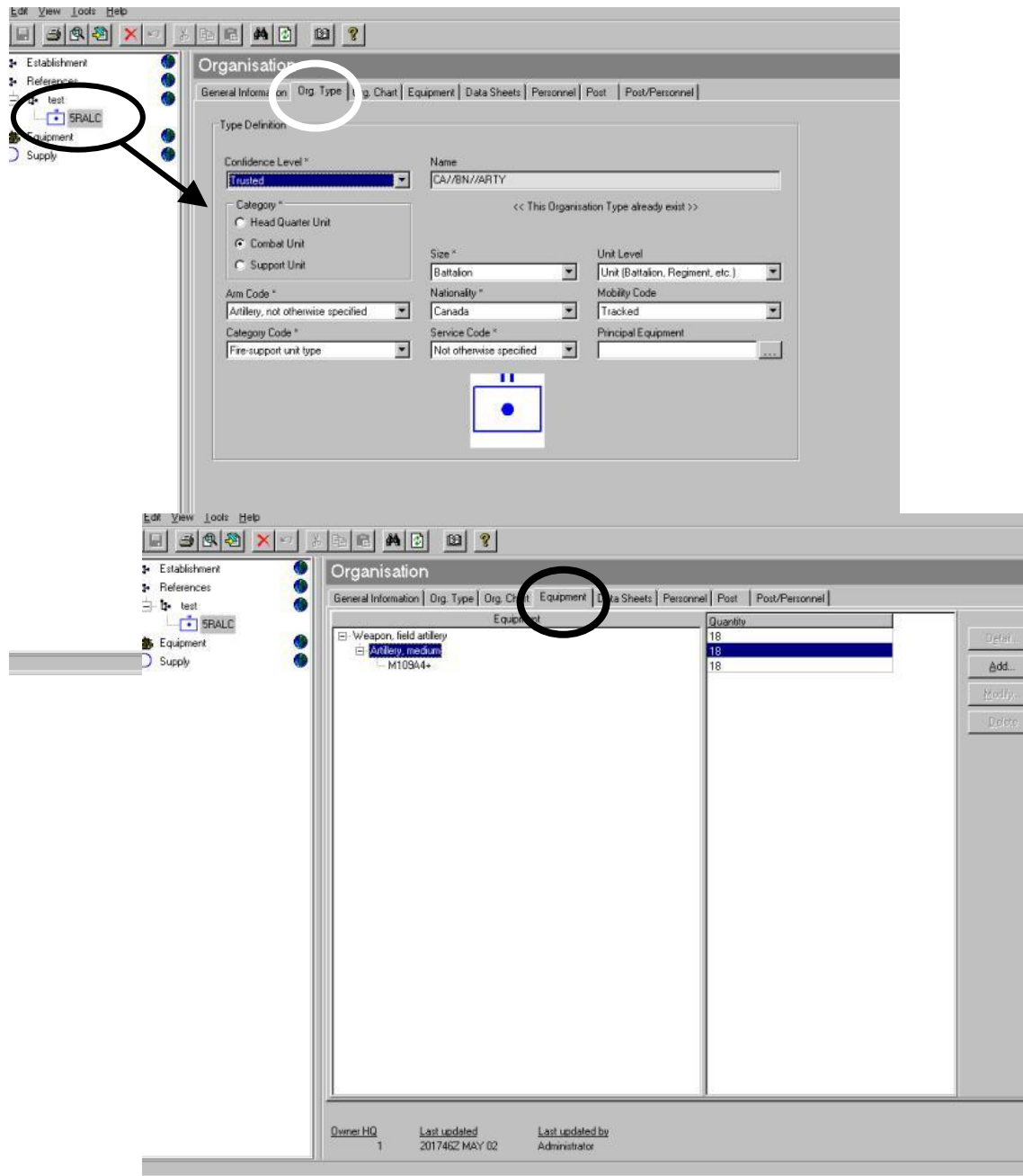


Figure 7-6 L'interface d'OPERA

### 7.2.3 La structure de la base de données

La structure de la base de données sous-jacente à OPERA est conforme au modèle de données de la Force terrestre (LFDM) basée sur le modèle de données LC2IEDM discuté au chapitre 6 – MilML – Un vocabulaire pour l'environnement militaire. Voici la liste des 10 entités clés indépendantes qui y avaient été mentionnées:

- OBJECT-ITEM
- OBJECT-TYPE
- CAPABILITY
- ACTION
- LOCATION
- CANDIDATE-TARGET-LIST
- REPORTING DATA
- CONTEXT
- REFERENCE
- RULE-OF-ENGAGEMENT

Le LFDM est un modèle relationnel entre les objets de la base de données dont voici une modélisation:

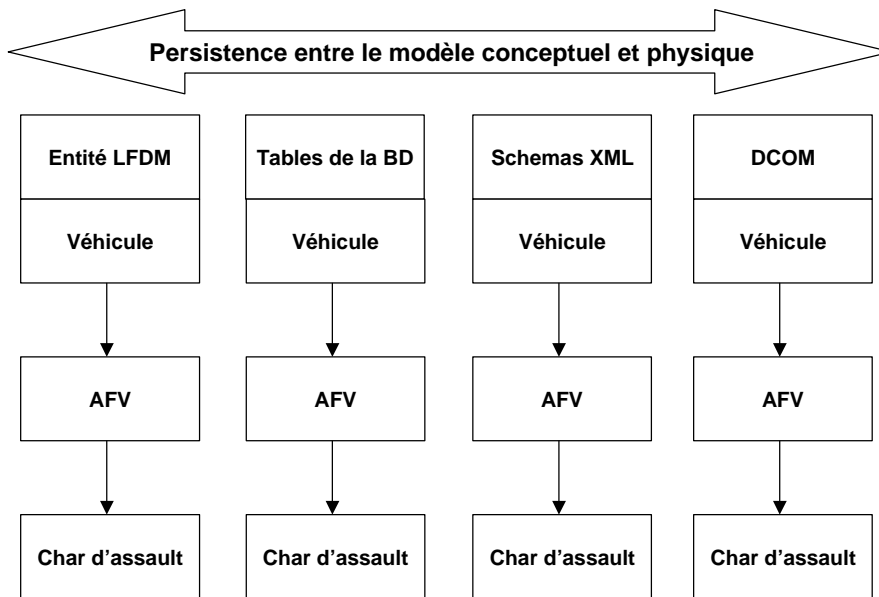


Figure 7-7 Le modèle relationnel orienté-objet de LFDM et OPERA (redessiné et adapté)<sup>8</sup>

Un véhicule est ici représenté dans sa forme générique et dans ses formes spécialisées, d'où la relation entre les objets. Un char d'assaut est un type de véhicule blindé de combat (AFV), de même qu'un AFV est un type de véhicule. On retrouve donc une relation père-fils qui est persistante dans tous les types d'implantation.

<sup>8</sup> Maj JR Couture, Weisman, Bob, "Army Integrated management Environment Phase 2 – Proof of Concept Brief", 1<sup>er</sup> mars 2000

### 7.2.4 Le flux de l'information

L'information dont a besoin OPERA pour fonctionner réside principalement dans l'ODS. Il y a donc un échange de données entre la base de données d'OPERA et l'ODS. Cet échange se fait à l'aide de deux technologies d'interopérabilité, soit DCOM et XML.

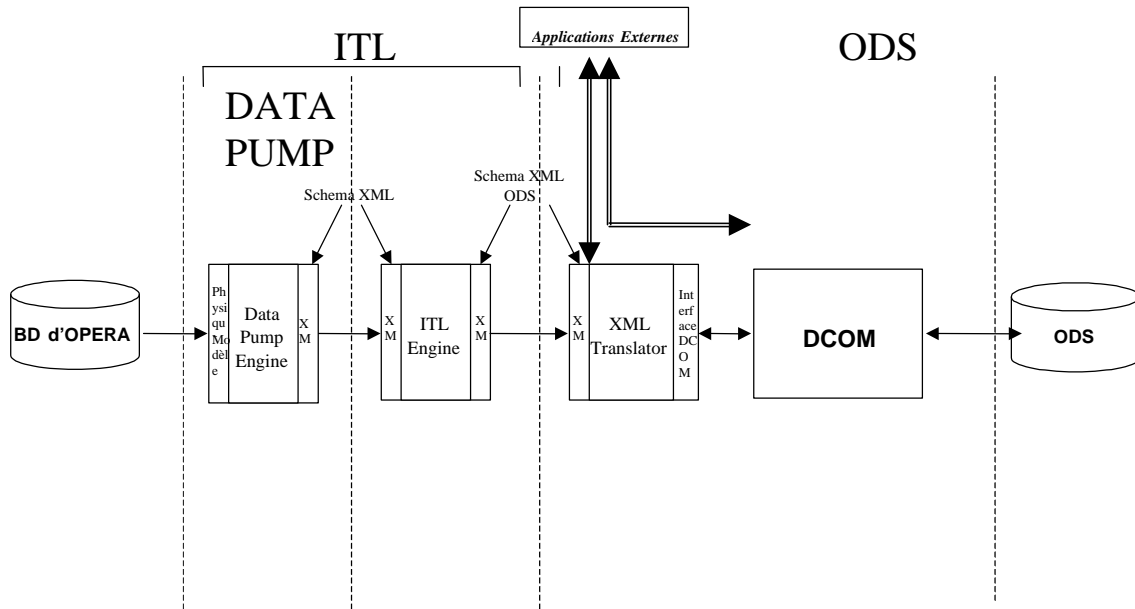


Figure 7-8 Le flux d'information au travers des couches de transfert de données (redessiné)<sup>9</sup>

La couche d'Information et de Transformation (ITL) extrait les données et produit un document XML basé sur un schéma XML. Il y a ensuite transformation du document XML en un autre basé cette fois sur un schéma XML d'ODS pour respecter cette nouvelle structure de base de données. Finalement un traducteur adapte le document XML pour que la technologie DCOM transporte l'information jusqu'à destination.

<sup>9</sup> Idem.

### 7.2.5 Extracteur de données (DATA PUMP)

L'extracteur de données contient du Meta-code afin de procéder à l'extraction des données à partir de la base de données d'OPERA. Il transforme ces données en format XML et fait suivre au gestionnaire de mise à jour ou dans un fichier XML. L'extracteur est lancé à l'aide d'un fichier XML contenant les paramètres décrivant la manière d'extraire les données et comment les faire suivre<sup>10</sup>

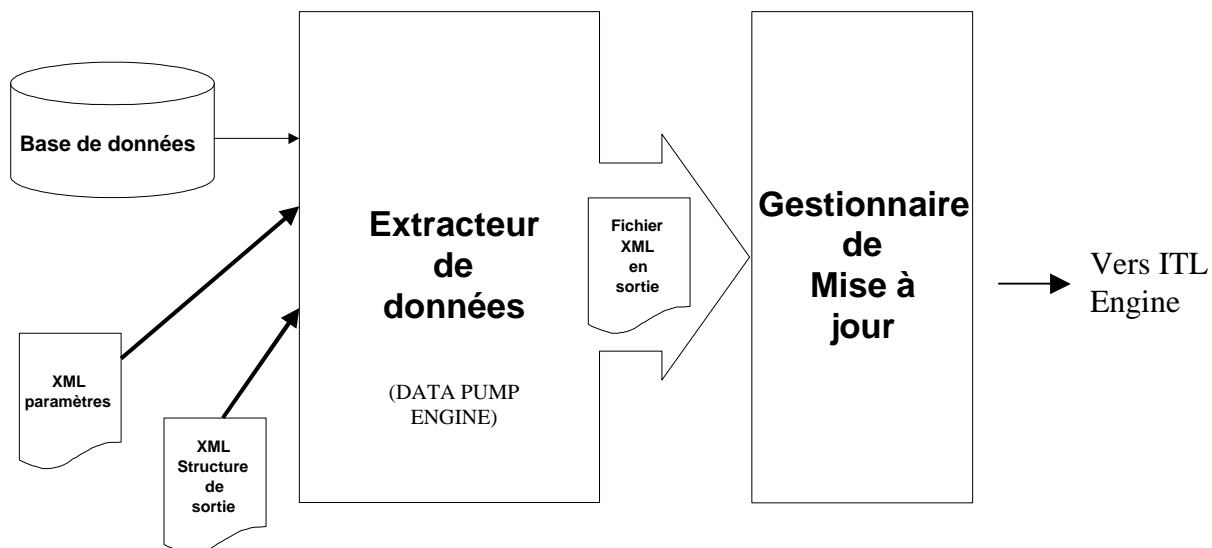


Figure 7-9 Fonctionnement de l'extracteur de données (redessiné)<sup>11</sup>

### 7.2.6 Le gestionnaire de mise à jour

Le but du gestionnaire de mise à jour est de s'assurer que l'information qui sera envoyée à l'ITL est nouvelle et ne contient pas d'anciennes données ayant déjà été transmises.

Puisque le gestionnaire de mise à jour fonctionne sur le principe de client-serveur, il utilise les services du système d'exploitation au démarrage de l'ordinateur. Il travaille en arrière plan et attend pour une connexion. Lorsqu'il reçoit l'information venant de l'extracteur, il envoie la différence de la dernière transmission à l'ITL<sup>12</sup>.

<sup>10</sup> Groupe CGI, "DATA PUMP Component Technical Architecture V2.2", 12 juillet 2000

<sup>11</sup> Idem.

<sup>12</sup> Groupe CGI, "Update Manager Technical Architecture V1.3", 28 juillet 2000

### 7.2.7 La couche d'information et de transformation (ITL)

L'ITL traduit les documents provenant de la structure du système source vers la structure de l'ODS. L'ITL reçoit un document du gestionnaire de mise à jour.

L'ITL fonctionne aussi comme un serveur et écoute les requêtes sur un "socket" TCP/IP en utilisant le protocole HTTP. Lorsqu'une connexion est établie, un script XML dirige les actions de l'ITL suivant un format de traduction. Un nouveau fichier XML est ainsi créé<sup>13</sup>.

---

<sup>13</sup> Groupe CGI, "Integration and Transformation Layer (ITL) Technical Architecture V1.2", 15 juillet 2000.



## 7.2.8 Le XML Transformer

Cette composante est utilisée par l'ITL durant la phase de traduction d'un document XML. Elle adapte le modèle des données sources au modèle de données de l'ODS pour le compte de l'ITL.

Le XML Transformer est le processus qui est appelé en premier par l'ITL car toutes autres modifications doivent être appliquées au modèle ODS<sup>14</sup>. Voici un exemple de transformation de fichier XML entre les deux modèles:

Fichier en entrée:

```
<ORGS from="AEMIS" to="ODS" subject="ESTABLISHMENT_ORGANISATION">
  <ORG>
    <FULL_NAME_E>12E REGIMENT BLINDE DU CANADA</FULL_NAME_E>
    <NAME_E>12 RBC</NAME_E>
    <ID>294</ID>
    <PARENT_ID>43</PARENT_ID>
    <UNIT_CODE>5246</UNIT_CODE>
    <SIZE_CODE>REGT</SIZE_CODE>
  </ORG>
  <ORG>
    ...
    ...
    ...
  </ORG>
</ORGS>
```

Fichier en sortie: (gabarit de mappage – "mapping template")

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LFDMDATA SYSTEM "lfdm.dtd">
<LFDMDATA>
  <Unit>
    <UnitId>ID</UnitId>
    <UnitFrmlAbrvdName>NAME</UnitFrmlAbrvdName>
    <UnitIdentCode>UNIT_CODE</unitIdentCode>
    <ObjtiName>FULL_NAME_E</ObjtiName>
  </Unit>
  <Unit>
    ...
    ...
    ...
  </Unit>
</LFDMDATA>
```

<sup>14</sup> Groupe CGI, "XML Transformer V 1.2", 17 juillet 2000

### 7.2.9 La synchronisation

En raison de plusieurs processus client-serveur prenant place au même moment, une synchronisation est essentielle pour éviter l'écriture par-dessus celle venant juste d'être mise à jour. Plusieurs processus d'extraction sont inter-dépendant et doivent attendre pour un autre processus d'extraction avant de finir afin d'utiliser sa sortie.

La synchronisation qui prend place est un mécanisme entre chaque instance de serveur<sup>15</sup>.

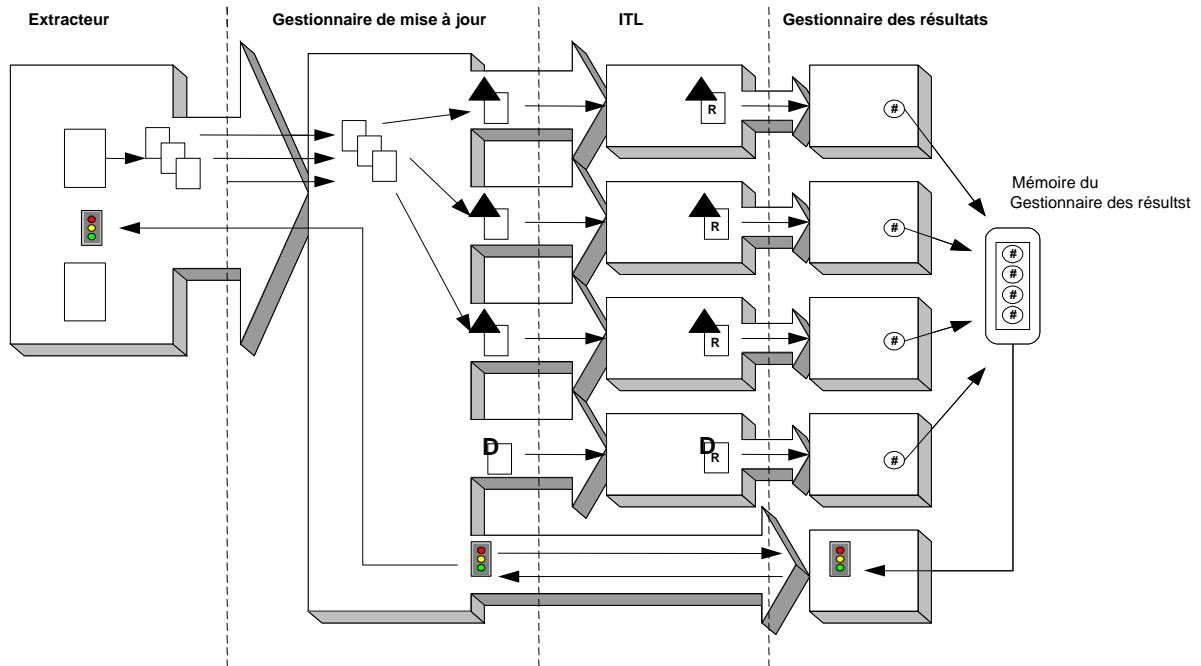


Figure 7-10 La synchronisation (redessiné)<sup>16</sup>

<sup>15</sup> Groupe CGI, "Synchronization Technical Architecture V1.2", 17 juillet 2000.

<sup>16</sup> idem

## 7.3 - Description du système d'information CMS

### 7.3.1 Généralités

Cette section cerner le contexte et les caractéristiques du Système de gestion des communications « CMS » qui sont pertinentes dans le contexte d'interopérabilité avec OPERA.

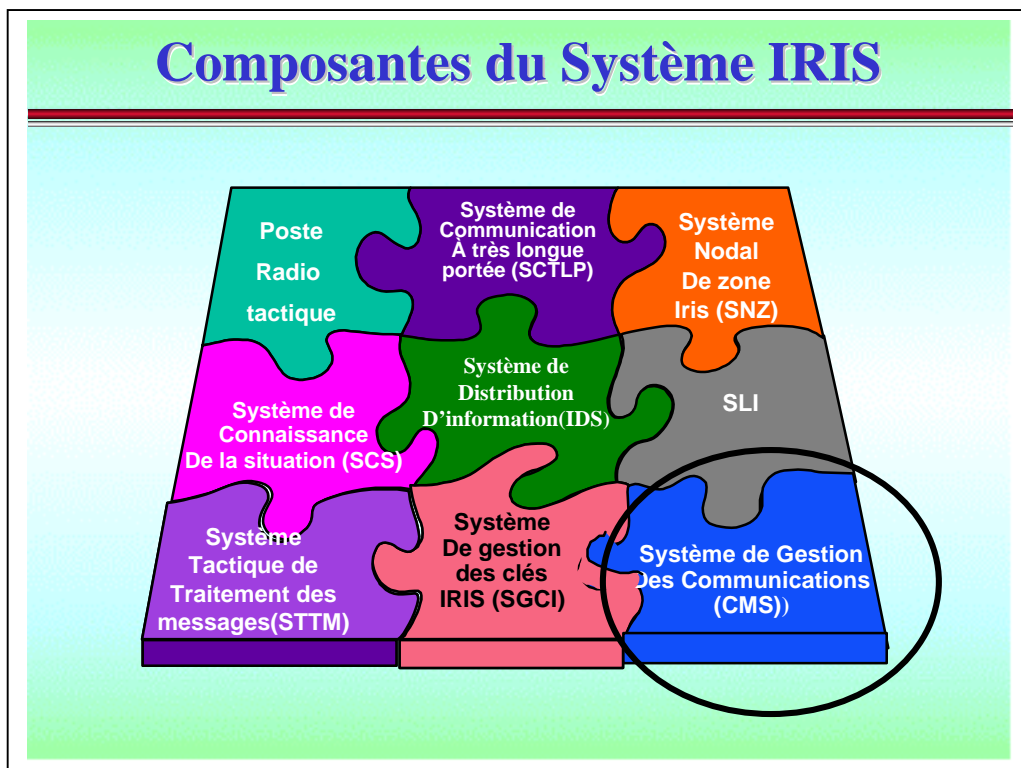


Figure 7-11 Composantes du Système IRIS (Traduit et redessiné)<sup>17</sup>

### 7.3.2 Le système IRIS

Le système IRIS offre aux organismes de transmission de l'Armée canadienne la possibilité d'exécuter des fonctions de gestion des communications entièrement transparentes aux éléments qu'ils appuient. Les composantes du système IRIS sont décrites à la figure 7-11. Un diagramme du contexte du Segment CMS est inclus à l'Annexe C. La composante d'intérêt principale est le système de gestion des communications « CMS ».

La gestion et l'opération du CMS respecte de près les voies hiérarchiques traditionnelles mais qui demeure entièrement au sein de l'organisme des transmissions<sup>18</sup>. Le CMS interagit sur les autres composantes du système IRIS en suivant la représentation de la figure 7-12. Une observation importante doit être faite en ce qui concerne

<sup>17</sup> Présentation BP STCCC, Briefing to LFTSP, 14 février 2002.

<sup>18</sup> BP STCCC/SLI, Aperçu du Système IRIS, C-53-750-000/MA-001, 26 mars 1999, page 4-2-1.

l'extension au système d'opération de l'anglais « Operating System Extension ». Dans ce service réseau « SDI », l'extension du système d'exploitation UNIX (IOSX) permet de dissimuler efficacement l'environnement UNIX pour l'opérateur. Ce système IOSX a été développé à l'aide du langage Ada et de logiciels du commerce<sup>19</sup>. Cette particularité est une embûche dans tout travail de développement qui repose sur l'accès direct avec le SCO OpenServer 5.0 d'Unix. C'est d'ailleurs la première observation faites par le gérant du CMS pour expliquer les défis du CMS.

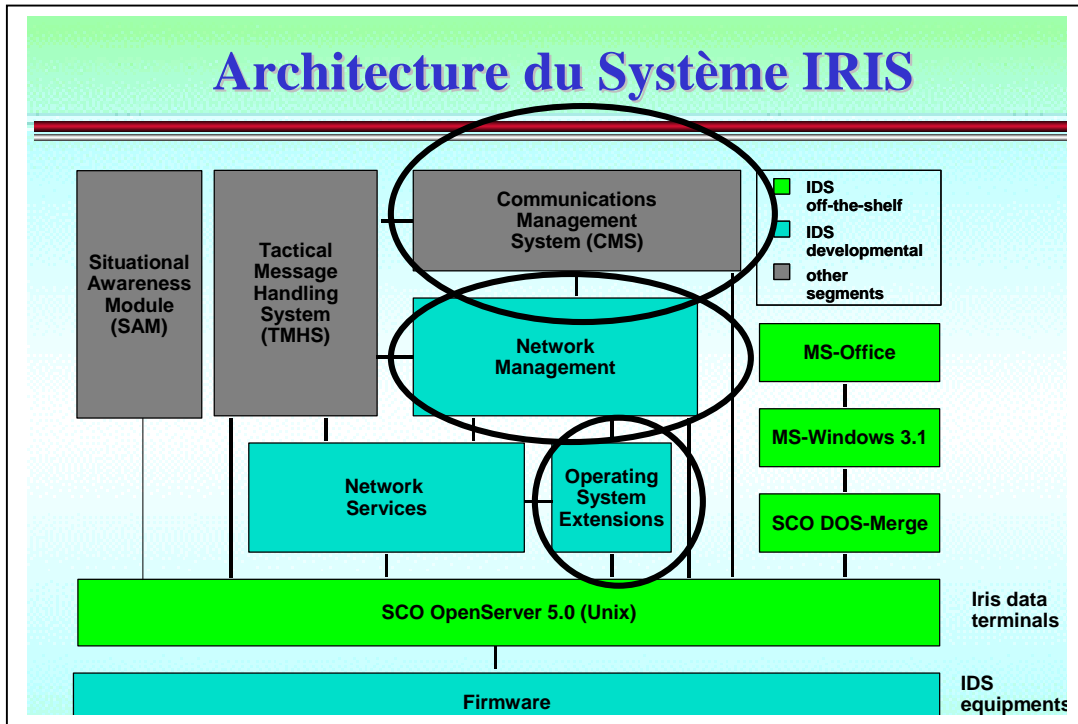


Figure 7-12 Architecture du Système IRIS<sup>20</sup>

### 7.3.3 Le système CMS

Le CMS se compose de deux applications distinctes mais physiquement intégrées dans le même poste nommé Terminal de Données Portatif (TDP) ou en anglais Portable Data Terminal « PDT » modélisé à la figure 7-13. Ces deux applications sont la planification et le contrôle des communications (PCC) en anglais « CPC », et la gestion du spectre des fréquences en ambiance tactique (GSFT) ou de l'anglais « BFSM ». L'échange de données entre ces deux systèmes est démontré à l'Annexe D et E. De ces deux applications, on tentera d'accéder l'élément PCC qui exige la capture de données manuelles.

<sup>19</sup> Idem, page 4-2-12.

<sup>20</sup> Présentation BP STCCC, Briefing to LFTSP, 14 février 2002.

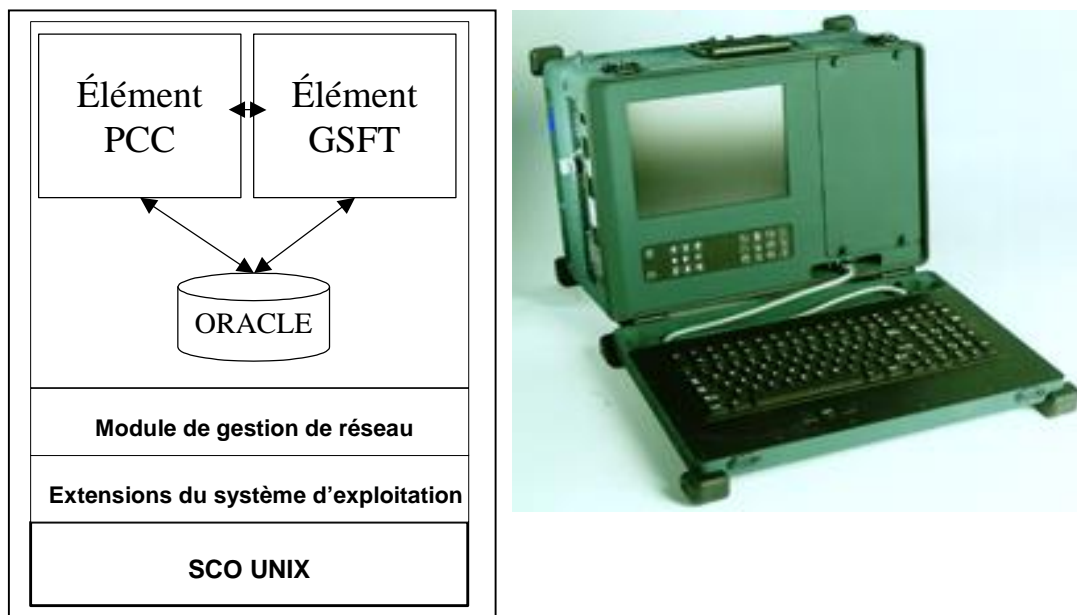


Figure 7-13 Terminal de données Portatif(TDP) (Photo TDP retirée de BP STCCC<sup>21</sup>)

L'élément PCC permet la planification, la coordination et le maintien d'une surveillance de l'état des ressources de communication affectées<sup>22</sup>. Le PCC offre à l'état-major des transmissions les services de logiciel nécessaire, de sorte qu'il puisse exécuter ce qui suit :

- a. Planifier le déploiement des communications, incluant la préparation électronique d'une série d'ordre et d'instruction;
- b. Mettre à jour les rapports d'état;
- c. Contrôler l'attribution des ressources de communication.

### 7.3.4 L'information capturée manuellement par CMS

En début de campagne, le contrôleur de ce système doit manuellement capturer la structure de l'unité pour lancer l'opération. L'hypothèse est que cette information est déjà numérisée dans OPERA, mais devra être répétée jusqu'à ce qu'une interface quelconque soit développée<sup>23</sup>. Cette tâche peut devenir fastidieuse car il faut passer par chacune des fenêtres (GUI) du système et confirmer l'information injectée. Ainsi pour l'unité 5 RALC, l'information suivante doit être répétée :

- a. Le numéro de plan de la campagne;

<sup>21</sup> Présentation BP STCCC, Briefing to LFTSP, 14 février 2002.

<sup>22</sup> BP STCCC/SLI, Aperçu du Système IRIS, C-53-750-000/MA-001, 26 mars 1999, page 4-2-5.

<sup>23</sup> Idem, page 4-4-1.

- b. Le nom de l'unité;
- c. L'organigramme de l'unité qui inclut le QG de l'unité, les batteries, les troupes, et les détachements;
- d. Le site de chacune de ses sous organisations;
- e. Les nœuds de communication; et
- f. Parmi les 140 véhicules, capturer la configuration de ceux ayant un émetteur radio, son indicatif d'appel, son rôle, l'identification du réseau radio.

Le titre des tables de la base de donnée CMS a été obtenu sous la forme d'un document Word. Les principales tables ont été reproduites à l'Annexe G. À l'aide de l'outil ERWIN et de l'assistance du BP STCCC, une approximation des relations entre chaque table a été déduite durant l'expérimentation. Le titre des tables ayant de l'intérêt pour ce projet sont :

- a. DEPLOYMENT\_PLAN;
- b. UNIT;
- c. SITE;
- d. NODE;
- e. VEHICLE;
- f. EMITTER; et
- g. ANTENNA.

### **7.3.5 Technologie utilisée dans l'environnement CMS**

Le système d'opération CMS repose sur une technologie maintenant ancienne encapsulée pour permettre la livraison d'un produit arrivé à terme. Son évolution technologique est terminée. La base de données est la version Oracle 7i pour laquelle le manufacturier a terminé le soutien technique. De même, le système d'opération SCO OpenServer 5.0.2 UNIX est devenu difficile à soutenir même au travers de General Dynamic Canada. Certaines fonctions ont aussi été programmées en utilisant le logiciel Ada adopté par le passé pour les applications militaires.

### **7.3.6 Effort requis pour l'interopérabilité**

L'interopérabilité est une activité d'actualité dans une vision du champ de bataille numérisé. Le CMS est un système ancien qui opère au sein du STCCC sans aucune autre interface automatisée. L'analyse de l'expérimentation doit démontrer des avantages dans l'élimination de recueillir de l'information déjà automatisée et dans la diminution des erreurs de frappe. Une des difficultés est de déterminer l'information et sa structure déjà disponible dans OPERA.

## **7.4 Modèle théorique d'interopérabilité OPERA/CMS**

### **7.4.1 Généralités**

Cette section vise à présenter, démontrer et expliquer le modèle théorique d'échange de données entre les deux applications OPERA et CMS.

La section 7.1 a déjà présenté la problématique existante au niveau de l'interopérabilité des deux systèmes en cause en soulignant la différence de structure et de système d'exploitation.

Voici les aspects couverts dans cette section:

- a. présentation des options;
- b. choix d'une option;
- c. sélection d'un langage; et
- d. analyse et design du programme.



## 7.4.2 Présentation des options

Nous avons identifié deux approches possibles afin de transférer des données du système OPERA vers le système CMS:

- a. Utilisation de la DATA PUMP d'OPERA; et
- b. Conception d'un injecteur de données.

### 7.4.2.1 Utilisation de la DATA PUMP d'OPERA

La base de données sous-jacente à OPERA doit fournir l'information de mise à jour à l'ODS. Cette tâche est effectuée grâce à la DATA PUMP d'OPERA qui procède à l'extraction et à la transformation de données dans un format compatible avec l'ODS.

Nous avons émis l'hypothèse qu'en utilisant la DATA PUMP d'OPERA (technologie Java), les données extraites pourraient être transformées selon une autre structure de base de données en utilisant un schema XML représentant la structure de CMS (gabarit de mappage). En raison du langage Java, la DATA PUMP peut être exécutée sur différents systèmes d'exploitation, ici WindowsNT et UNIX.

La figure suivante illustre ce concept en dérivant le concept de la figure 7-6. La base de données ODS est ainsi remplacée par celle sous-jacente au CMS et une deuxième DATA PUMP installée sur la plateforme UNIX s'occupe de l'insertion des données au lieu de faire appel à la technologie DCOM, puisque le CMS n'est pas en réseau.

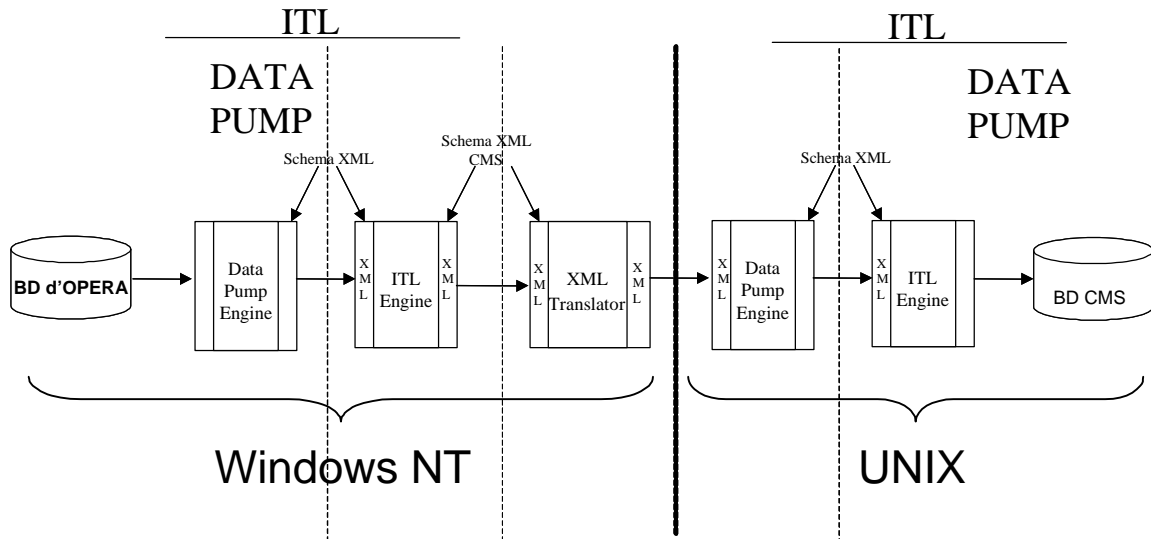


Figure 7-14 Concept d'utilisation de la DATA PUMP Java d'OPERA avec CMS

En lançant l'exécution de la DATA PUMP d'OPERA qui prend ses paramètres des schemas XML pré-existants, un fichier XML qui respecte la structure de la base de données de CMS est produit. Après avoir introduit ce fichier du côté de la plateforme UNIX, on lance l'exécution de la DATA PUMP du CMS, qui lit et interprète le document XML et insère les données dans la base de données CMS.

#### 7.4.2.2 Conception d'un injecteur de données

Au lieu d'utiliser la DATA PUMP d'OPERA pour injecter l'information dans la base de données sous-jacente à CMS, on conçoit notre propre application qui prendra en entrée un fichier XML fournit par la DATA PUMP d'OPERA et qui effectuera la lecture et l'insertion des données dans la base de données de CMS.

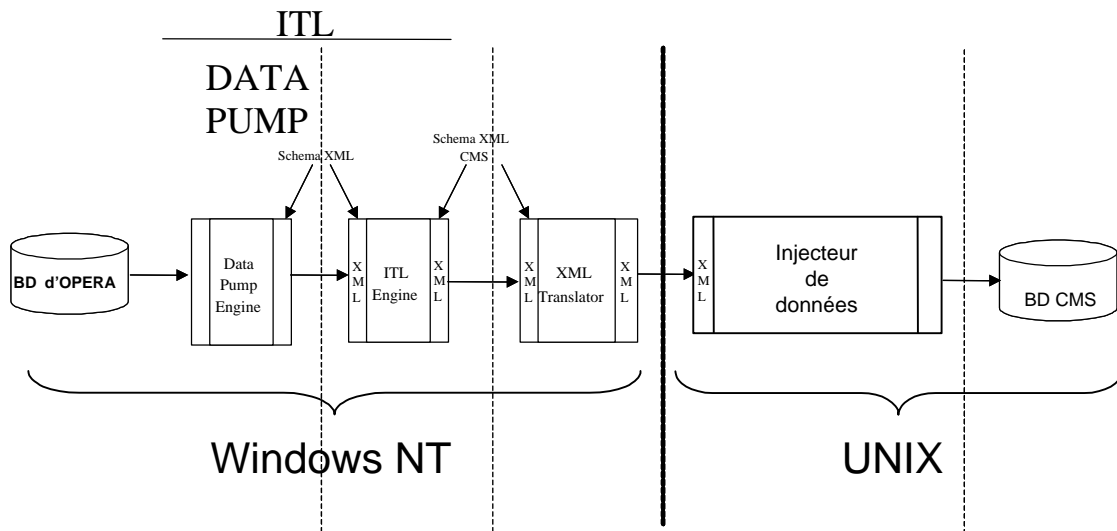


Figure 7-15 L'injecteur de données XML

L'injecteur de données est une application simple et portable, pouvant fonctionner sur différents systèmes d'exploitation. Lors de son exécution, elle établit une connexion avec la base de données de CMS, et procède à la lecture du document XML en entrée.

Parcourant chaque noeud de l'arborescence XML, elle établit par programmation la correspondance entre la source et la destination et finalement insert la donnée dans la base de données.

### 7.4.3 Choix d'une option

Nous avons choisi de concevoir notre propre injecteur XML plutôt que d'utiliser la DATA PUMP d'OPERA afin de se connecter à la base de données de CMS et d'y insérer l'information nécessaire.

La raison majeure justifiant ce choix est l'architecture fermée du système IRIS et de CMS. L'accès au code source de CMS est géré par GD Canada et certaines méthodes et procédures reliées à la saisie des données dans l'interface utilisateur de CMS nous sont inconnues et inaccessibles. Pour fonctionner, la DATA PUMP d'OPERA devrait être adaptée et nous pensons qu'il vaut mieux dans ce cas partir à neuf. Une nouvelle application pourra mieux intégrer les actions à prendre lorsque certains champs de la base de données CMS sont modifiés.

Voici la démarche à suivre afin de concevoir cette application:

- a. identifier un langage de programmation flexible et portable. La capacité d'exécuter cette application sur plusieurs plate-formes différentes est un atout;
- b. procéder à l'analyse des tables des bases de données des systèmes OPERA et CMS afin de déterminer les correspondances désirées et les différences d'architecture;
- c. construire l'application en s'assurant d'incorporer les mises à jour sur les champs de tables dépendantes ainsi que l'incrémentaire requise pour la synchronisation.

### 7.4.4 Sélection d'un langage

Le langage sélectionné doit rencontrer certains critères:

- a. il doit pouvoir interpréter un document XML à l'aide d'un analyseur(parseur) XML;
- b. il doit être exécutable sur les deux systèmes d'exploitation suivants:
  - (1) Windows NT; et
  - (2) UNIX;
- c. il doit permettre une communication avec la base de données de CMS.

Les langages qui se sont avérés correspondre le mieux à nos critères sont entre autres, le langage Java, Perl, et PHP.

Le PHP a été sélectionné en raison de sa simplicité<sup>24</sup>. C'est un langage de script, qui permet donc une lecture plus accessible du code source. Il est distribué gratuitement et est plus portable que Java. Le PHP a un analyseur XML et le support pour ORACLE 7 intégrés.

#### 7.4.5. Design du programme

Les concepts suivants serviront de base pour l'expérimentation. La réalisation de ce programme confirmera nos hypothèses sur l'interopérabilité entre les systèmes OPERA et CMS.

##### 7.4.5.1 Commentaires de spécification de l'injecteur de données

Voici les commentaires de spécification devant servir de point de départ à notre application.

Injecteur de données

```
{  
Objectif:      À partir d'un document XML, insérer des données dans la base de données  
               de CMS  
Méthode:      Lire le document XML et se connecter à la base de données pour insérer  
               l'information  
Besoin:       Un document XML valide et conforme à la structure de la base de données  
               de CMS  
Entrée:       Un document XML  
Sortie:       Données à insérer  
Résultat:     --  
Hypothèse:    Le document XML fourni est valide et conforme à la structure de la base  
               de données CMS tel que produit par la DATA PUMP d'OPERA  
}
```

---

<sup>24</sup> Kolesnikowicz, Jeff, "XML Application for CMS/Opera Compatibility", livré le 6 juin 2002

#### 7.4.5.2. Pseudo-Code de l'application

Dans cette phase de conception, voici en pseudo-code les plus importantes instructions de l'application injecteur de données:

```
ouvrir le fichier XML, le transcrire dans un DOM XML
Tant qu'il y a des noeuds à lire
  Lire le premier noeud
  Tant qu'il y a des enfants sous le noeud
    Lire la balise du premier enfant
    Rechercher la base de données pour un champ correspondant
    Insérer les données de la balise dans l'espace mémoire approprié
    Mettre à jour l'enregistrement
    Vérifier si une opération doit être conduite sur un autre champ ou table
    Si oui
      Effectuer l'opération
  Enfant suivant
Noeud suivant
```

La première étape consiste à ouvrir le fichier XML fourni en entrée et à le placer en mémoire sous forme arborescente selon le concept de Document Object Model (DOM). Tant qu'il y a des items à lire dans le fichier, l'application lira l'arbre ainsi formé et effectuera la correspondance entre la balise et les champs de la base de données. Lorsqu'il y a correspondance, l'information contenue dans la balise est insérée à l'endroit approprié de la base de données et cette dernière est mise à jour.

Finalement, l'application effectue une vérification selon les instructions incluses. Si de par la nature de la nouvelle donnée injectée, une opération affectant un autre champ dans la base de données est nécessaire, telle l'incrémentement d'un compteur ou la mise à jour d'un champ de vérification de temps, l'application effectue cette opération.

#### 7.4.6. L'analyse des tables des systèmes OPERA et CMS

Les données à échanger entre les bases de données d'OPERA et de CMS étant conservées dans des tables, nous nous attarderons quelque peu à analyser ces dernières. La liste complète des tables OPERA et CMS est à l'annexe F et G.

##### 7.4.6.1 Analyse des tables de CMS

En partant du besoin, c'est donc avec CMS que débutera cette analyse. Le but de notre application est de fournir à CMS une partie des données requises pour remplir les tables de la base de données de CMS.

Parmi les tables de CMS, nous avons identifié celles devant recevoir des données lors de la saisie de nouvelles entrées. Sommairement, toute activité dans la base de données de CMS débute par la création d'un nouveau plan de déploiement (deployment plan) auquel est assigné un numéro de plan. Une unité est ensuite créée, avec une combinaison de sous-unités, et de véhicules sont attachés à ces sous-unités.

Un site est relié à une unité et peut aussi recevoir une série de véhicules. Le même principe s'applique à une node.

Voici une représentation relationnelle de ces tables:

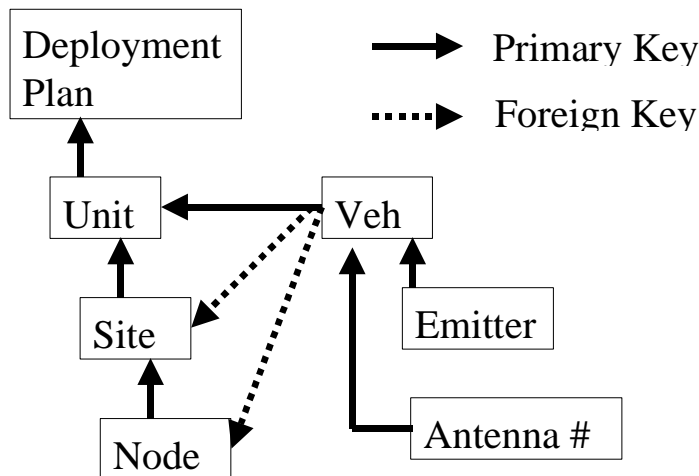


Figure 7-16 Représentation relationnelle des tables de CMS (redessiné)<sup>25</sup>

Lors de la saisie de données dans CMS, des clés séquentielles (keys) sont attribuées, tels PLAN\_KEY, UNIT\_KEY, SITE\_KEY, VEHICLE\_KEY, etc... Ce sont ces clés qui permettent le lien entre les tables et leurs éléments.

<sup>25</sup> Clement, Capt Jason, BP STCCC, Courriel "FWD SCO Compiler for CMS", 15 mai 2002.

Voici un exemple d'une de ces tables:

(CMS) VEHICLE

	VEHICLE_KEY: NUMBER(6)
	UNIT_KEY: NUMBER(6)
	SITE_KEY: NUMBER(6)
	ICON_NAME: VARCHAR2(45)
	ICON_TYPE: NUMBER(2)
	VEHICLE_TYPE_NAME: VARCHAR2(20)
	VEHICLE_ID: VARCHAR2(40)
	NODE_KEY: NUMBER(6)
	VEHICLE_CALLSIGN: VARCHAR2(4)
	TEMP_USED: NUMBER(1)

Tableau 7-1 Table VEHICLE de CMS (tiré de l'annexe G)

Les termes NUMBER et VARCHAR2 du tableau 7-1, aussi présents dans toutes les tables de CMS signifient le type de données devant être utilisé, suivis du nombre de caractères autorisés entre parenthèses. NUMBER indique un nombre et VARCHAR indique une chaîne de caractères.

#### 7.4.6.2 Analyse des tables d'OPERA

La structure de la base de données d'OPERA est totalement différente de celle de CMS. Elle est beaucoup plus "éclatée". Sa forme normale (FN) a été optimisée pour donner une excellente intégrité référentielle des données. Elle est bâtit en suivant exactement le modèle hiérarchique des objets du LC2IEDM vu dans un chapitre précédent.

La structure particulière des tables d'OPERA en fait une force au niveau de la performance, mais rend son analyse plus difficile, spécifiquement afin de faire correspondre les tables des deux systèmes d'information.

Nous avons néanmoins cerné quelques tables du système OPERA qui devraient contenir ou tout au moins nous diriger vers les données recherchées. Il s'agit des tables physiques:

- a. EQPTT;
- b. EQPTT\_DTL;
- c. UNITT;
- d. ORBAT;

- e. ORG;
- f. UNIT;
- g. OMATA;
- h. HOLDING;
- i. MAT;
- j. ORG\_S; et
- k. MAT\_S.

Les tables EQPTT et EQPTT\_DTL sont le point de départ pour le matériel. Entre autre, un véhicule pourra être retrouver à l'aide du code ECC que l'on retrouve dans ces tables. Les tables UNITT, UNIT, ORG contiennent l'information sur les unités, alors que la table ORBAT nous donne le lien hiérarchique qui existe entre elles. La table HOLDING contient de l'information sur la quantité de chacune des ressources. Les tables ORG\_S et MAT\_S donnent un aperçu des unités et du matériel en évolution dans le temps. La table OMATA établit l'association entre les organisations et le matériel.

Voici un exemple des tables UNIT d'OPERA et de CMS en comparaison

(NATO) UNIT - UNIT	(CMS) UNIT
UNIT-ID (FK)	UNIT_KEY: NUMBER(6)
UNIT-FORMAL-ABBREVIATED-NAME	PLAN_KEY: NUMBER(6)
UNIT-IDENTIFICATION-CODE	ICON_NAME: VARCHAR2(45)
UNIT_DISPLAY_ORDER: numeric(6)	ICON_TYPE: NUMBER(2)
UNIT_IS_HQ_FLAG: varchar(20)	UNIT_NAME: VARCHAR2(14)
UNIT_PARENT_AFFILIATION_FLAG : varchar(20)	PARENT_UNIT: NUMBER(6)
UNIT_PARENT_COUNTRY_FLAG : varchar(20)	
UNIT_SPECIAL_SIZE_FLAG: varchar(20)	
UNIT_LAST_UPDATE_DTG: datetime	
OWNER_ID: varchar(20)	
UPDATE_SEQNR: numeric(18)	
LAST_UPDATED: datetime	
LAST_UPDATED_BY: varchar(30)	
UNIT_INST_QTY: numeric(3)	
REFRESH_DATE: datetime	
OWNER_HQ: numeric(5)	

Tableau 7-2 Table UNIT d'OPERA et tableau 7-3 Table UNIT de CMS



On peut voir dans ces deux tables qu'une unité autant dans OPERA que dans CMS possède un numéro d'identification. UNIT-ID dans OPERA et UNIT\_KEY dans CMS. Si ces deux numéros ne correspondent pas entre eux, notre application devra posséder un dictionnaire de correspondance pour effectuer la traduction entre les deux systèmes. Des exemples similaires existent dans chacune des tables impliquées.

### 7.4.6.3 Le mappage et la correspondance des tables

L'expérimentation devra examiner la correspondance entre les tables d'OPERA et de CMS afin d'identifier les données communes et significatives. Cette correspondance est essentielle au partage d'information et le seul fait de ne pas utiliser un modèle commun augmente la difficulté.

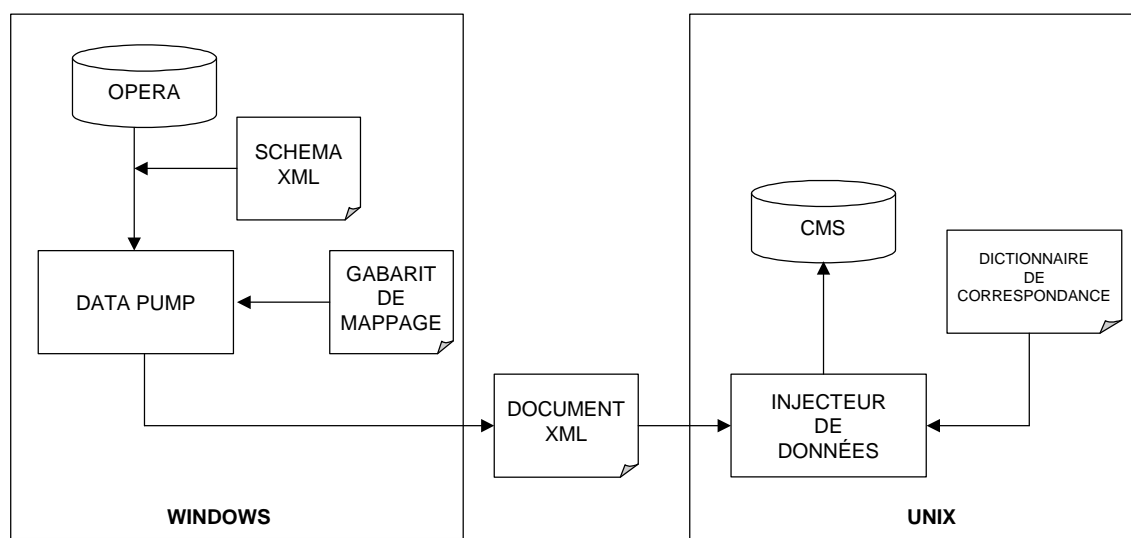


Figure 7-17 Effectuer la correspondance entre les tables des deux systèmes

La figure 7-17 illustre une solution afin d'assurer une correspondance des champs et de l'information qu'ils contiennent. Le schema XML indique à la DATA PUMP quelle information extraire de la base de données. Ce schema XML est basé sur l'information correspondant à la structure d'OPERA qui doit être transférée dans CMS.

Le gabarit de mappage (mapping template) donne au XML Transformer la manière de structurer le fichier XML qui sera produit afin qu'il corresponde cette fois à la structure de CMS. On parle ici de la structure des balises qui entourent les données à transférer.

Finalement, l'injecteur de données se sert du dictionnaire de correspondance pour insérer les données importées d'OPERA et non conformes à celle devant être injectées dans CMS. On parle ici des données se trouvant entre les balises du document XML.

Le schema, le gabarit de mappage et le dictionnaire de correspondance sont tous des documents XML eux-mêmes.

## **7.5 Expérimentation avec le modèle théorique**

### **7.5.1 Généralités**

Jusqu'à présent toute la recherche s'est fait de façon théorique en obtenant et consultant toute la documentation technique disponible. La démarche d'expérimentation consistera à décrire le matériel d'expérimentation obtenu pour le banc d'essai et les efforts d'expérimentation avec chacune des plateformes. On y inclura une description des problèmes encourus en particulier la difficulté d'obtenir du soutien technique pour des applications reposant sur une technologie désuète.

### **7.5.2 Description du banc d'essai**

Le matériel de base a été réuni pour créer un banc d'essai au sein du Département de science militaire appliquée situé à Kingston. Ce simple banc d'essai nécessite deux ordinateurs PC non reliés. Le premier ordinateur sera configuré pour contenir le système d'information OPERA et le deuxième l'application CMS. Ces deux ordinateurs ont été obtenus de la cellule d'entraînement STCCC située à Kingston en février 2002.

Pour OPERA, le besoin initial a été rencontré avec un ordinateur IBM, Personnel Computer 300 GL, Pentium II, 350 MHz, 128 MG PC-100 SD RAM – disque dur de 8.4 GB SCSI ainsi que le système d'opération Windows NT 4.0. Un nouvel ordinateur aurait été nécessaire pour rencontrer le besoin de la nouvelle version d'OPERA.

En ce qui concerne la plateforme CMS, il a été impossible de libérer un Terminal de Données Portatif (TDP) de campagne, qui sont présentement en distribution au sein des forces terrestres. Toutefois, la cellule d'entraînement pour le STCCC a libéré un TDP B ayant des caractéristiques plus performantes que la version de campagne permettant ainsi d'améliorer la vitesse de traitement lors des essais. Le TDP B est un Compaq Deskpro, Pentium III, 450 MHz, 128 MG PC-100 SD RAM – disque dur 91 GB ultrawide 2 SCSI.

En sommaire, le banc d'essai comprenait initialement des ordinateurs ayant les caractéristiques souhaitables pour l'expérimentation. Toutefois, pour parer aux problèmes déjà encourus, l'équipe du Groupe CGI a changé la configuration d'OPERA déclassant ainsi le PC IBM. Tandis que pour CMS, le PC Compaq est considéré comme une version plus performante que le système distribué en campagne

### 7.5.3 Expérimentation avec OPERA

En ce qui concerne la plateforme OPERA, les besoins en configuration matérielle ont changé bien après la rencontre initiale avec l'équipe de CGI<sup>26</sup>. Dans la première installation d'OPERA, le besoin minimal identifié était un serveur fonctionnant sur le système d'opération Windows NT 4 – Service Pack 5. Une disponibilité d'au moins 1.5 GB de mémoire morte ainsi que l'installation Microsoft SQL Server 7.0. De même, pour la création d'une nouvelle base de donnée OPERA, au moins 1.2 GB devait être libéré<sup>27</sup>.

Ayant éprouvé de la difficulté avec la DATA PUMP, notre équipe du Département de science militaire appliquée a tenté d'obtenir du soutien technique de l'équipe de CGI. C'est à ce moment que l'on apprenait que l'équipe de CGI a fait face à des problèmes de configuration des modules CORBA les forçant à abandonner leur effort de développement avec cette première configuration d'OPERA. La technologie DCOM serait maintenant utilisée. Cette information n'est parvenue à Kingston que le 24 avril 2002<sup>28</sup>. Cette version beta d'OPERA a besoin d'une plateforme plus puissante. Tout d'abord l'application repose sur un système d'exploitation Windows NT 4 SP6 Serveur ou Windows 2000 Serveur. La configuration physique de la plateforme recommandée est un Pentium III de 853 MHz, ayant 512 MG RAM et 40 Gig de disque dur<sup>29</sup>. Cette information devenait trop tardive pour obtenir un ordinateur possédant les caractéristiques souhaitables. Une timide tentative d'expérimentation a été infructueuse avec la plateforme IBM précédent.

### 7.5.4 Expérimentation avec CMS

Une courte session d'entraînement a été fournie par M. Rick Russell de la cellule d'entraînement du STCCC à Kingston. Le matériel de référence est un aide-mémoire expliquant les étapes à suivre pour faire fonctionner CMS<sup>30</sup>. Cette session a fournie une indication du temps requis pour remplir complètement un organigramme d'une unité, qui inclut les sous-unités, le plan de référence, les véhicules, les sites d'opération, les nœuds de transmission et les émetteurs radio. Il a été estimé qu'environ huit à dix heures de capture de données pourrait être requis pour remplir complètement ce travail pour une unité.

La prochaine étape, était d'accéder à la base de donnée du CMS pour obtenir l'information requise sur la structure des champs normalement capturés par l'opérateur des transmissions. Cette étape est devenue un problème majeur dans l'expérimentation pour lequel du soutien technique sur le système d'opération UNIX était requis. Idéalement, on s'attend à ce que le manufacturier, soit General Dynamic Canada (GD

---

<sup>26</sup> Rencontre à Québec avec M. Richard D'Anjou/Ms Marie Théberge CGI, 8 novembre 2001.

<sup>27</sup> CGI et Centre de Recherche de la Défense de Valcartier, Opera Installation V 1.2, 13 novembre 2001, page 3.

<sup>28</sup> Théberge, Marie, CGI, Courriel OPERA, 24 avril 2002.

<sup>29</sup> CGI et Centre de Recherche de la Défense de Valcartier, Opera Installation Guide (Draft), version 0.1, 14 avril 2002, page 2.

<sup>30</sup> Lt. N. Pedneault & Lt. B. Cornell, 2 CMBG TCCCS Technical Coordination Cell, Communication Management Aide-Mémoire, sans date, 49 pages.

Canada), puisse fournir ce soutien technique, mais le gérant de l'équipement CMS a estimé que ce soutien mobiliserait trop de ressources de GD Canada qui doit déjà s'affairer à des tâches prioritaires pour le ministère. Pour parer à cette restriction, l'autorisation d'engager un spécialiste ayant de l'expérience en génie du logiciel a été obtenue donc le rapport est à l'annexe GG.

La première constatation d'importance lors de la période d'expérimentation est l'absence d'outils de développement SCO UNIX dans la version installée de ce système d'exploitation. Il est présumé que la version SCO UNIX du CMS est une installation sur mesure développée pour une plateforme de capacité d'entreposage restreinte ayant un besoin de sécurité accru comme le TDP. Il a été impossible d'obtenir une copie originale du disque d'installation SCO UNIX 5.0.2. du gérant de l'équipement<sup>31</sup> néanmoins des outils de développement âgés sont disponible sur le site Internet de Caldera<sup>32</sup> qui distribue UNIX.

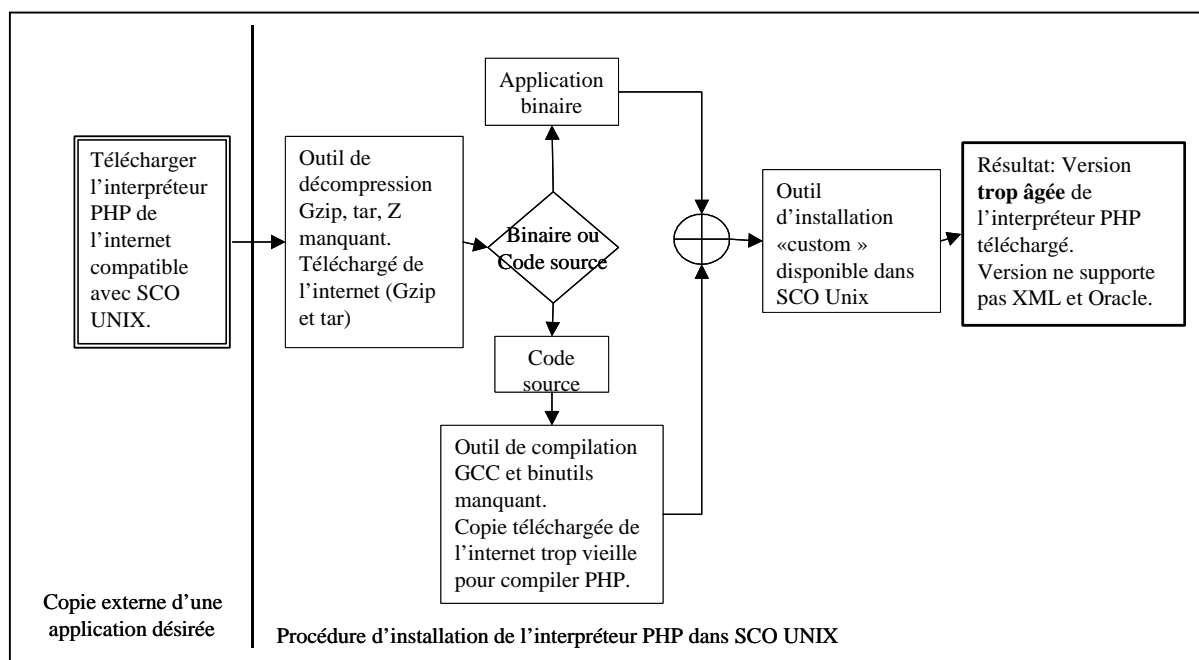


Figure 7-18 Processus de chargement d'application propre à SCO UNIX (Redessiné à partir d'XML Application for CMS/Opera Compatibility<sup>33</sup>)

Avant d'entreprendre tout travail sur la base de donnée, il devenait essentiel de munir le système UNIX avec l'interpréteur PHP sélectionné pour entamer le développement avec XML. L'expérimentation décrite à la figure 7-18 a permis d'établir que l'outil interpréteur PHP, les outils de décompression « Gzip » et « tar », et les outils de compilation « GCC » et « binutils » sont tous manquants dans le SCO UNIX de CMS.

<sup>31</sup> Watts, John, DLCSPM 2-4, SCO Compiler for CMS, 14 May 2002.

<sup>32</sup> <<http://www.caldera.com/skunkware>>

<sup>33</sup> Kolesnikowicz, Jeff, XML Application for CMS/Opera Compatibility, livré le 6 juin 2002, page 9.

Seul l'outil d'installation « custom » est disponible. Après diverse recherche d'outil sur l'internet<sup>34</sup>, il a été réalisé que la copie PHP qui fonctionne est trop âgée pour supporter XML et ORACLE. À l'opposé, les outils « GCC » et « binutils » sont incapables de compiler une version moderne de PHP pouvant soutenir du développement avec XML et ORACLE. Cette impasse ne pourra être franchit qu'en engageant des efforts de développement additionnels pour moderniser les outils « GCC » et « binutils ». Cette approche pourrait occasionner des coûts très exorbitants dont les bénéfices anticipés sont modestes. Cette évaluation dépasse largement le cadre de cette expérimentation.

---

<sup>34</sup> Idem, pages 8-11.

## 7.6 Recommandations liées à l'interopérabilité entre OPERA et CMS

Cette section est consacrée à l'aspect interopérabilité entre les deux applications OPERA et CMS. Quelques mots à propos du contexte dans lequel s'est déroulé cette expérimentation. D'un côté une application en développement, ayant des défauts connus, et de l'autre une application en livrée avec une architecture fermée. L'hypothèse de départ était de démontrer l'interopérabilité entre les deux systèmes. Malgré les difficultés encourues les recommandations énoncées ici aideront à établir qu'il est en fait possible de rendre les deux systèmes interopérables, toutefois ils doivent évoluer vers de nouvelles versions.

L'information recueillie par l'expérimentation met en lumière les possibilités mais aussi les difficultés liées à l'atteinte de l'interopérabilité à grande échelle, surtout en ce qui concerne de nombreux systèmes d'information disparates.

Deux recommandations majeures et une série d'observations mineures ressortent de l'expérimentation. Voici les deux recommandations majeures afin d'accomplir l'interopérabilité entre OPERA et CMS:

- a. **Moderniser CMS.** Mettre à jour le système d'exploitation qui supporte CMS vers un système d'opération plus performant comme Windows NT, Windows 2000 ou LINUX qui devrait harmoniser les communications et les outils de développement avec OPERA. Les ressources matérielles telles le processeur et la mémoire devront probablement être mis à jour en même temps afin de faire fonctionner ce système d'exploitation et les applications sous-jacentes; et
- b. **Harmoniser le modèle de données.** La différence entre le modèle du LC2IEDM et celui de CMS est si grande que les efforts de correspondance mettent en péril l'interopérabilité entre les deux systèmes. Une révision complète du modèle de données de CMS pourra amener des changements bénéfiques. De plus, la version ORACLE de la base de données supportant CMS doit aussi être mise à jour.

Voici les observations et recommandations mineures entre OPERA et CMS:

- a. **Validité de XML comme outil d'interopérabilité.**

Les caractéristiques du XML sont bien adaptées pour résoudre le genre de problématique à laquelle on fait face ici avec OPERA et CMS. La simplicité du XML et son utilisation de balises significatives à l'intérieur d'un format texte normalisé sont sa force comme outil d'interopérabilité.

- b. **Nouvelle technologie vs vieille technologie.**

Il a été constaté que le XML a tout de même ses limites si les systèmes auxquels on tente de l'y adapter utilisent une technologie dépassée. C'est le

cas ici de CMS qui utilise SCO UNIX, un système d'exploitation qui ne possède pas la technologie pour supporter le XML et dont aucun développement n'est effectué par l'industrie. Si on tente d'établir des liens d'interopérabilité avec d'autres systèmes semblables, le risque d'échec n'est pas négligeable. L'évolution rapide de la technologie est une caractéristique principale de l'informatique et nos systèmes doivent suivre cette évolution pour rester interopérables entre eux.

c. **Documentation sur les modèles de données.**

Un outil qui aurait été de grande valeur afin d'établir la correspondance entre les tables d'OPERA et de CMS est une documentation précise des deux modèles de données impliqués. La tendance est d'établir sous forme graphique les modèles de données. Les développeurs d'un système en particulier peuvent sans doute s'y retrouver mais pour quelqu'un de l'extérieur, il devient très difficile de comprendre les détails du modèle. Ce genre de documentation devrait fournir l'information sur chacune des tables et ses relations avec les autres, une description détaillée de la signification de chacun des termes utilisés dans la nomenclature des noms de champs. Les champs qui sont utilisés et ceux qui ne le sont pas. La disponibilité d'un outil semblable économisera des heures voir des jours de recherche à tenter de faire correspondre les deux bases de données.

d. **Expansion du modèle de données.**

Il a été remarqué que certains champs comme l'indicatif d'appel qu'utilise intensément CMS est inexistant dans OPERA. Cela nous amène à nous interroger sur la possibilité pour la base de données d'OPERA de prendre de l'expansion et d'inclure des champs qui sont utilisés par d'autres systèmes afin d'établir une base commune d'information à laquelle des systèmes externes peuvent accéder et ainsi réduire la redondance entre les systèmes d'information.

e. **Utilisation des Schemas XML.**

La DATA PUMP utilise des DTD pour définir les documents XML qu'elle produit. Il est recommandé l'utilisation de Schemas XML plutôt que des DTD. Les DTD ne sont pas écrits en syntaxe XML et ne peuvent pas être lus aussi facilement qu'un document XML par un analyseur XML. Le typage de données serait également mieux pris en charge par les Schemas puisque une DTD n'utilise que le type caractère.

## **Chapitre 8 – Conclusion et Recommandations**

### **8.0 Conclusions**

#### **8.1 Généralités**

Les nouvelles technologies amènent toujours des changements. Cette déclaration est autant vraie pour notre société que pour le domaine militaire dans lequel nous évoluons. Pour les militaires, le changement apporté par les nouvelles technologies se traduit par le développement de nouvelles méthodes de travail, voir un changement de doctrine. Nos soldats réalisent que malgré la présence d'outils informatisés, leurs tâches n'ont pas diminué et qu'ils doivent quand même capturer la même information pour desservir plusieurs SI indépendants au sein d'une brigade. La structure de nos forces et de notre organisation pourrait changer plus rapidement si l'interopérabilité était complètement réalisée. Nos efforts de développement et d'acquisition d'équipement doivent adopter une approche de systèmes de systèmes et faire de l'interopérabilité une priorité pour les SI des FT.

Les FT traversent une période de changement axée sur l'intégration des systèmes d'information dans les systèmes d'armes et dans les processus de commandement et de contrôle. Beaucoup d'effort est mis sur l'exploitation des possibilités des nouvelles technologies afin d'accomplir l'interopérabilité des systèmes d'information entre alliés et à l'interne.

Il ne suffit pas que d'apprendre un langage comme XML pour assurer l'interopérabilité entre deux SI différents. Dans ce cas-ci, obtenir les ressources nécessaires pour l'expérimentation n'était qu'une difficulté. Il a fallu se familiariser avec deux systèmes d'exploitation différents (Windows NT et UNIX), deux bases de données différentes (SQL Server et ORACLE), deux SI (OPERA et CMS), un logiciel de soutien ERWIN pour déterminer et représenter la nature des relations au sein de chacune des bases de données, être prêt à faire fonctionner un analyseur (XML), bâtir un outil d'interopérabilité (Injecteur de données), et finalement comprendre le modèle de données LFD des FT puis faire correspondre les champs des tables de données semblables. Après avoir compris l'ensemble des tâches énumérées, chacune d'elle peut devenir un problème incontournable si l'assistance technique n'est pas disponible.

#### **8.2 Solutions technologiques aux problèmes d'interopérabilité**

Des solutions existent aux problèmes d'interopérabilité grâce à des technologies diverses qui agissent comme pont entre les systèmes d'information. Les technologies d'interopérabilité discutées sont le CORBA, le COM et DCOM, le RMI et le XML. Ce qui est devenu évident à la suite de ce projet est la complexité d'implantation de certaines technologies ainsi que l'expertise requise pour en assurer le fonctionnement à l'exception d'XML.



### 8.3 Le potentiel du XML

Le XML a sans contredit un énorme potentiel pour atteindre l'interopérabilité des systèmes d'information des Forces terrestres. Il doit cependant être utilisé de concert avec d'autres technologies selon la situation et la nature du problème rencontré. Son principal avantage par rapport à ses rivaux est sa simplicité et sa facilité d'implantation.

Le langage XML est un langage simple basé sur des fichiers textes. Sa force est sa capacité de meta-langage, c'est-à-dire sa capacité à transmettre de l'information au sujet d'information. Il décrit l'information qu'il contient, permettant de donner le contexte et de transmettre un vocabulaire expliquant comment traiter les données. Cette caractéristique du XML le rend tout à fait adapté à l'échange d'information entre applications informatiques. Il permet de faire le lien entre des systèmes incompatibles pour autant que son format puisse être interprété par l'application qui tente de le lire.

Le XML permet de traiter avec aisance tous les types de données et même d'en créer de nouveaux à l'aide des schémas XML. Il permet d'éviter le conflit entre des données semblables lors d'échange d'information à l'aide des espaces de nommage (namespaces). Le XML peut même invoquer à distance des applications afin de faire la requête automatique d'information et d'en obtenir des réponses, grâce au Simple Object Access Protocol (SOAP).

Sa normalisation établie par le World Wide Web Consortium en fait une spécification fiable à implanter et assure son interopérabilité parmi ses propres composantes. L'industrie du commerce électronique reconnaît d'emblée son potentiel pour l'échange d'information et stimule des progrès constants au niveau des capacités et de performances d'interopérabilité du XML. Cela se traduit par une facilité à obtenir de l'expertise, contrairement à des langages qui ne sont pas supportés par la communauté informatique (comme le langage Ada).

Le domaine militaire a lui aussi reconnu le potentiel certain du XML par la création d'un vocabulaire XML propre au domaine militaire, le MilML. Ce vocabulaire permet aux nations alliées qui l'adoptent de communiquer selon une structure et un langage commun. Le MilML est un vocabulaire axé sur le soutien des opérations tactiques qui définit et traite les informations opérationnelles et administratives. Le modèle de données développé par l'OTAN (LC2EIDM) et utilisant le MilML comme vocabulaire assurera une interopérabilité lors de l'utilisation des futurs systèmes d'information de commandement et de contrôle construits sur ce modèle.

#### **8.4 L'aspect sécurité**

Des progrès récents dans le développement des composantes XML apportent une composante novatrice au niveau de la sécurité de l'information. Cet aspect est d'une importance primordiale lorsqu'on parle d'interopérabilité entre alliés sur le champ de bataille. Le XML est maintenant capable de rendre des services d'authentification, d'intégrité, de non répudiation, de confidentialité et d'habilitation. Utilisées de concert, les technologies XML Signature, XML Encryption et "Security Assertion Markup Language" (SAML) fournissent un environnement sécuritaire à l'utilisation du XML pour l'échange de données entre des applications réseau. Ces technologies supportent la notion d'un intranet sécuritaire à l'échelle de l'OTAN.

#### **8.5 L'expérimentation**

Cette expérimentation touchant deux systèmes d'informations de la FT, soit OPERA et CMS, a mis à jour les défis à relever pour atteindre l'interopérabilité. Malgré que la preuve n'ait pas été faite sur la solution axée sur le XML, elle est censée et réalisable. Toutefois, la différence de technologie existant entre les deux systèmes est si grande que l'implantation physique de cette solution dépassait le cadre du projet et aurait nécessité des efforts supplémentaires.

Les conclusions sur CMS sont nombreuses. Premièrement, le système d'opération est à architecture fermée. Il a été impossible de pénétrer le système et de rejoindre la base de données ORACLE. Donc l'injection de données venant d'un système externe à Iris n'a pas été possible dans le cadre du projet. En plus CMS ne bénéficie pas de soutien technique en provenance du manufacturier General Dynamic Canada ou même pour le SCO UNIX qui est discontinué.

En ce qui concerne OPERA, la DATA PUMP qui extrait des données du système OPERA utilise des DTD pour définir les documents XML. Les DTD ne peuvent pas prendre en charge les types de données et les espaces de nommage, augmentant les efforts de développement en raison des vérifications supplémentaires en programmation à réaliser.

La différence d'architecture au niveau des deux bases de données impliquées ont aussi augmenté la difficulté, et a demandé le besoin d'une longue analyse détaillée de leur structure respectives.

Les difficultés rencontrées jettent néanmoins une lumière sur des difficultés semblables qui peuvent survenir si on tente d'implanter des technologies d'interopérabilité sur une plus grande échelle. Les SI existants qui reposent sur d'anciennes technologies devront être modifiés avant de pouvoir les inclure dans un effort d'interopérabilité.

## **8.6 Recommandations**

Le chapitre 7 contient des recommandations spécifiques au problème d'interopérabilité entre les applications OPERA et CMS. On y énonce deux recommandations majeures et cinq recommandations mineures. Il est valable de s'attarder aux deux recommandations majeures.

### **8.6.1 Utilisation du XML en interopérabilité**

Il est fortement recommandé d'évaluer la possibilité d'adopter le XML comme outil principal en interopérabilité pour les systèmes d'information des FT. Le projet démontre le potentiel du XML et sa facilité de développement et d'implantation.

Une recherche plus approfondie devrait être menée afin de déterminer si SOAP pourrait être utilisé afin d'établir des appels de méthode distants et ainsi automatiser bon nombre d'échange d'information considéré répétitifs. Tout SI des FT fonctionnant dans un contexte réseau pourrait bénéficier de la technologie SOAP.

Il est recommandé de maximiser l'utilisation des schémas XML comme la norme pour définir des documents XML. L'effet bénéfique de cette utilisation devrait se faire sentir par des efforts moindres en développement d'application et en programmation puisque le type de données transporté par un document XML n'aurait plus à être traité par programmation grâce au schéma XML.

Au niveau de l'intégrité référentielle des données, l'utilisation des schémas XML en concert avec les espaces de nommage permettrait d'éviter les conflits lors de collision de balises semblables. Ce point est particulièrement important dans un contexte d'interopérabilité au niveau de l'OTAN.

Il est recommandé que les technologies XML traitant de sécurité soient étudiées afin d'envisager leur utilisation intégrée à de futurs SI. Le projet a démontré le potentiel du XML en sécurisation de transfert de documents par réseau. Il y a lieu de penser que les efforts en implantation d'un tel système pourraient sauver des ressources en raison de la simplicité du XML, on suggère ainsi l'interopérabilité des systèmes de sécurité.

### **8.6.2 Modèles de données**

Il est recommandé que le modèle de données LFDM serve de modèle aux SI qui devront être interopérables et qu'une analyse visant à harmoniser leur modèle de données respectives soit conduite. Ceci pour tenter de minimiser les efforts d'interopérabilité en raison de la présence potentielle de modèles de données trop différents qui empêcherait un échange efficace d'information. Cet effort d'harmonisation pouvant être considérable, nous suggérons qu'il se limite aux données communes qui feraient l'objet d'un échange électronique.

La documentation décrivant les modèles de données devrait être révisée afin de s'assurer d'une plus grande clarté et de plus de détails sur chacun des éléments et entités composant le modèle. Une des plus grandes difficultés de la recherche fut de comprendre le modèle de données.

### **8.6.3 Modernisation des systèmes anciens**

Les systèmes d'information utilisant d'anciennes technologies de bases de données et des systèmes d'exploitation dépassés devraient être modernisés avant de procéder à des efforts d'interopérabilité.

### **8.6.4 Contrôle de l'information**

Un aspect de l'implantation qui nous laisse sans réponse est la responsabilité de contrôle de l'information. Une fois les systèmes d'information interopérables, qui aura la responsabilité de contrôler l'information nouvellement partagée? Alors qu'auparavant différentes entités revendiquaient la propriété et la mise à jour d'une information dans des systèmes indépendants, un réseau de systèmes interopérables ne tolérera pas des informations ayant plusieurs propriétaires. Cela soulève un dilemme qui se doit d'être résolu. Il est donc recommandé que l'armée de terre se penche sur cette question afin qu'une politique soit établie.

## Annexe A<sup>1</sup> - Convention Logique-à-physique des Noms

La table suivante représente les abréviations utilisées dans la modélisation de données MilML Schema. La “Source” indique l’origine des abréviations.

Nom Logique	Nom Physique	Source
abbreviated	abrvd	DLCI
absolute	abs	ATCCIS
access	acces	ATCCIS
accelerated	accel	DLCI
accelerated-stop	acstop	DLCI
account	acct	ATCCIS
accuracy	acc	ATCCIS
action	action	DLCI
action-functional	afunc	DLCI
action-functional-association	afunca	DLCI
action-aircraft-employment	aacem	DLCI
action-context	actxt	DLCI
action-temporal-association	atempoa	DLCI
action-detail	adtl	DLCI
action-effect	aeff	DLCI
action-event	aevent	DLCI
action-objective	aobj	DLCI
action-objective-item	aobji	DLCI
action-objective-type	aobjt	DLCI
action-resource	ares	DLCI
action-resource-employment	aresem	DLCI
action-resource-item	aresi	DLCI
action-resource-type	arest	DLCI
action-required-capability	areqtcapab	DLCI
action-task	atask	DLCI
activity	activ	ATCCIS
address	addr	ATCCIS
administrative	admin	ATCCIS
administrative-feature	afeat	DLCI
administrative-feature-type	afeatt	DLCI
aircraft	acrft	ATCCIS
airfield	airfld	DLCI
allotment	allot	ATCCIS
alternate	alt	ATCCIS
altitude	altid	DLCI
amount	amnt	ATCCIS
angle	angle	ATCCIS
answer	ans	ATCCIS
approach	aprch	ATCCIS
approval	apprv	ATCCIS
approved	apprd	ATCCIS
area	area	ATCCIS
assessing	assg	ATCCIS
assessment	assmt	ATCCIS
assignment	asgmt	DLCI

<sup>1</sup> CGI, Military Mark up Language - A proposal for Military Interoperability Language based on XML and the NATO LC2IEM, draft version 1.7, 31 mai 2000, Annexe A.

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
association	a	DLCI
atmosphere	atm	ATCCIS
attack	atk	ATCCIS
absolute-timing	abstimg	DLCI
precision-code	pr_code	DLCI
authorisation	auth	ATCCIS
authorising	authg	ATCCIS
authority	authy	ATCCIS
average	avg	ATCCIS
availability	aval	DLCI
awards	awrd	DLCI
battlespace	btlsp	DLCI
bearing	bear	ATCCIS
beacon	becn	DLCI
bill	bill	DLCI
billable	billb	DLCI
birth	birth	ATCCIS
blood	blood	ATCCIS
bounding	bndg	ATCCIS
bridge	br	DLCI
budget	bgt	DLCI
canadian	ca	ATCCIS
candidate	cand	ATCCIS
candidate-target-detail	ctgtdet	ATCCIS
candidate-target-detail-association	ctgtdeta	DLCI
candidate-target-detail-item	ctgtdeti	DLCI
candidate-target-detail-type	ctgtdett	DLCI
candidate-target-list	ctgtlst	ATCCIS
candidate-target-list-association	ctglsta	DLCI
capability	capab	ATCCIS
capacity	capac	ATCCIS
category	cat	ATCCIS
city	city	ATCCIS
civic	cvc	ATCCIS
civil	civ	ATCCIS
classification	cl	ATCCIS
ceiling	ceil	DLCI
clearance	clrce	ATCCIS
cloud-cover	cldcovr	ATCCIS
cloud-layer	clidlayr	DLCI
code-value	codeval	DLCI
code	code	ATCCIS
combat	cbt	ATCCIS
common	com	ATCCIS
communication	comm	ATCCIS
completion	compl	ATCCIS
component	cmpnt	ATCCIS
composed	compd	ATCCIS
compound	cmpnd	ATCCIS
common-picture	cop	DLCI
computer	compu	ATCCIS
condition	cond	ATCCIS
conduct	cndc	ATCCIS

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
conduct-sheet	cdsh	DLCI
cone	cone	ATCCIS
confirmation	cfm	ATCCIS
consumable-materiel-type	consmatt	DLCI
contact	contct	DLCI
context	ctxt	ATCCIS
contract	ctract	ATCCIS
control	ctrl	ATCCIS
controled	ctrld	ATCCIS
control-feature	cfeat	ATCCIS
control-feature-control-feature-association-status-control-feature-control-feature-association	cccfeatasa	DLCI
control-feature-control-feature-association	ccfeata	DLCI
control-feature-control-feature-association-status	ccfeatas	DLCI
control-feature-geographic-feature-association-status-control-feature-geographic-feature-association	cgcgfeatasa	DLCI
control-feature-geographic-feature-association-status	cgfeatas	DLCI
control-feature-geographic-feature-association	cgfeata	DLCI
convoy	cvy	DLCI
coordinate	coord	ATCCIS
cost	cost	DLCI
count	cnt	ATCCIS
country	cntry	ATCCIS
coverage	cov	ATCCIS
creation	creat	ATCCIS
credibility	cred	ATCCIS
credit	cr	ATCCIS
crossing	cross	DLCI
criticality	critc	ATCCIS
currency	cur	ATCCIS
datatype	dtype	ATCCIS
date	date	ATCCIS
day	day	ATCCIS
debit	dt	DLCI
default	deft	ATCCIS
definition	defn	ATCCIS
degree	degr	ATCCIS
dependency	depcy	ATCCIS
description	descr	ATCCIS
detail	dtl	ATCCIS
dimension	dim	ATCCIS
distance	dist	DLCI
direction	dir	ATCCIS
discriminator	disc	ATCCIS
displaced	displd	DLCI
distribution	distr	ATCCIS
document	doc	ATCCIS
domain	dom	ATCCIS
driver	dvr	ATCCIS

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
dummy	dum	ATCCIS
duration	dur	ATCCIS
duty	duty	ATCCIS
easting	east	ATCCIS
economical	econo	ATCCIS
effect	eff	ATCCIS
effective	efftv	ATCCIS
effective-time	etime	DLCI
egress	egrs	ATCCIS
element	elm	ATCCIS
elementary	elmt	ATCCIS
elevated	elevd	ATCCIS
elevation	elev	ATCCIS
employment	empmt	ATCCIS
end	end	ATCCIS
engagement	engag	ATCCIS
engineering	eng	ATCCIS
environment	env	ATCCIS
equipment	eqpt	ATCCIS
equipment-type	eqptt	DLCI
equivalent	eqv	ATCCIS
established-organisation-type	estborgt	DLCI
established-materiel-type	estbmatt	DLCI
establishment	estb	ATCCIS
estimated	est	ATCCIS
ethnic	ethnc	DLCI
evaluation	eval	ATCCIS
evaluator	evalr	ATCCIS
event	event	ATCCIS
exception	excep	ATCCIS
exchange	xchg	ATCCIS
facility	fac	ATCCIS
facility-type	fact	DLCI
facility-facility-association	ffaca	DLCI
facility-component	fcmpnt	DLCI
facility-component-association	fcmpnta	DLCI
facility-feature-association-status- facility-feature-association	fffffeatasa	DLCI
facility-feature-association	ffeata	DLCI
fan	fan	ATCCIS
feature	feat	ATCCIS
feature-type	featt	DLCI
financial	finl	DLCI
financial-account-financial-account- association	fafinacca	DLCI
financial-account	finacc	DLCI
financial-period-budget	finlpdbgt	DLCI
financial-time-period	finltimepd	DLCI
financial-transaction	fintrn	DLCI
flight	flt	DLCI
fire	fire	ATCCIS
forecasted	forec	DLCI
formal	frml	ATCCIS



Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
format	fmt	ATCCIS
fraction	frac	ATCCIS
functional	func	ATCCIS
gap	gap	ATCCIS
gender	gdr	ATCCIS
geographic	geo	ATCCIS
geographic-feature	gfeat	ATCCIS
geometric	geom	ATCCIS
gradient	grad	ATCCIS
graphic	gra	ATCCIS
guidance	guid	ATCCIS
guiding	guidg	ATCCIS
halt	halt	ATCCIS
headquarters	hq	ATCCIS
height	hgt	ATCCIS
holding	hold	ATCCIS
honours	hon	ATCCIS
horizontal	horiz	ATCCIS
hostile	hos	ATCCIS
hostility	hostly	ATCCIS
humidity	hum	ATCCIS
id	id	ATCCIS
identification	ident	DLCI
identifiant	id	ATCCIS
immediate	immed	ATCCIS
index	idx	ATCCIS
indicator	ind	ATCCIS
inflight	inflt	ATCCIS
inflight-report	iflrp	ATCCIS
intelligence	int	ATCCIS
interest	intr	ATCCIS
interpretation	interp	ATCCIS
issuing	issg	ATCCIS
jane-break	jbrk	DLCI
land	land	ATCCIS
landing	lndg	DLCI
land-manoeuvre	lmanov	DLCI
language	lang	ATCCIS
* latitude	lat	ATCCIS
length	lht	ATCCIS
level	lvl	DLCI
light	lit	DLCI
load	ld	ATCCIS
lowest	lowst	DLCI
* location	loc	ATCCIS
logical	logal	ATCCIS
* longitude	long	ATCCIS
lower	low	DLCI
low-level-wind-shear	llws	DLCI
magnetic	magn	DLCI
major-part	main	DLCI
manoeuvre	manov	ATCCIS
marking	mkg	ATCCIS

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
mastered	mastr	ATCCIS
material	mat	ATCCIS
materiel	mat	ATCCIS
materiel-materiel-association	mmata	DLCI
materiel-type-materiel-type-association	mattmatta	DLCI
materiel-type	matt	ATCCIS
materiel-type-detail	mtdtl	DLCI
materiel-materiel-type-establishment	mmattestb	DLCI
materiel-type-establishment	mattestb	DLCI
matricule	cfid	ATCCIS
matrimonial	matrm	DLCI
minimal-altitude	mialti	DLCI
maximal-altitude	mxalti	DLCI
maximum	max	ATCCIS
measure	meas	ATCCIS
media	media	ATCCIS
message	msg	ATCCIS
meteorologic	met	ATCCIS
meteorologic-feature	mfeat	DLCI
method	meth	ATCCIS
military	mil	ATCCIS
military-load-classification	mlc	ATCCIS
mine	mn	ATCCIS
minefield	mnfd	ATCCIS
minimum	minm	DLCI
mission	missn	DLCI
mobility	mob	ATCCIS
mode	mode	ATCCIS
moon	moon	ATCCIS
name	name	ATCCIS
nation	natn	DLCI
national-language-support	nls	DLCI
nationality	naty	DLCI
nature	nat	ATCCIS
network	net	ATCCIS
nickname	nn	DLCI
night	ni	DLCI
* node	node	ATCCIS
nonperson	npers	ATCCIS
norm	norm	ATCCIS
nothing	nortg	DLCI
number	nbr	ATCCIS
object	ob	DLCI
object-item	oi	DLCI
object-item-association	oia	DLCI
object-item-association-status	oias	DLCI
objective	obj	ATCCIS
object-type	ot	DLCI
object-type-establishment	otestb	DLCI
object-type-detail-part	otdtlpa	DLCI
object-type-detail	otdtl	DLCI
observation	obsn	ATCCIS

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
occurrence	occur	ATCCIS
offset	offset	ATCCIS
one	one	ATCCIS
operation	op	ATCCIS
operational	oper	ATCCIS
* organisation	org	ATCCIS
organisation-organisation-association-status	oorgas	DLCI
organisation-organisation-association	oorga	DLCI
organisation-control-feature-association	ocfeata	DLCI
organisation-control-feature-association-status-organisation-control-feature-association	ococfeatas	DLCI
organisation-materiel-association	omata	DLCI
organisation-organisation-type-association	oorgta	DLCI
organisation-organisation-type-establishment	oorgtestb	DLCI
organisation-person-association	opersa	DLCI
organisation-type	orgt	ATCCIS
organisation-type-establishment	orgtestb	DLCI
* organisation-type-establishment-materiel-type-detail	orgtestb_mtdtl	DLCI
* organisation-type-establishment-organisation-type-detail	orgtestb_otdtl	DLCI
organisation-type-establishment-person-type-detail	orgtestb_ptdtl	DLCI
orientation	orien	ATCCIS
origin	ori	DLCI
original	orig	ATCCIS
output	outp	ATCCIS
overrun	ovrrn	DLCI
owner	owner	ATCCIS
packet	pack	ATCCIS
parent	parnt	ATCCIS
part	pa	ATCCIS
participating	prtcg	ATCCIS
participation	prtcn	ATCCIS
pattern	pat	ATCCIS
perception	percp	ATCCIS
period	pd	ATCCIS
perishability	peris	ATCCIS
persistance	persis	ATCCIS
person	persn	ATCCIS
personal	persnl	ATCCIS
personnel	pers	ATCCIS
person-person-association	ppersna	DLCI
person-type	perst	ATCCIS
phase	ph	ATCCIS
physical	phy	ATCCIS
picture	pict	ATCCIS
planned	pland	ATCCIS
planning	plang	ATCCIS
point	pt	ATCCIS

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
post	post	ATCCIS
post-type	postt	DLCI
postal	postl	ATCCIS
precipitation	prcp	ATCCIS
precision	pr	ATCCIS
pressure	pres	ATCCIS
primary	prim	ATCCIS
principal	princ	ATCCIS
priority	pri	ATCCIS
probability	prob	ATCCIS
proficiency	prof	ATCCIS
progress	prog	ATCCIS
province	prov	DLCI
purpose	purp	ATCCIS
qualification	qual	ATCCIS
qualifier	qualf	ATCCIS
quantity	qty	ATCCIS
range	rge	ATCCIS
rate	rate	ATCCIS
reception	recep	ATCCIS
manager	mgr	DLCI
reference	ref	ATCCIS
refraction	refr	ATCCIS
region	regn	ATCCIS
regular	reg	ATCCIS
reinforcement	rft	ATCCIS
relation	reln	ATCCIS
relative	rel	ATCCIS
religion	relgn	ATCCIS
report	rept	ATCCIS
reporting	reptg	ATCCIS
reporting-data	rptd	ATCCIS
request	req	ATCCIS
required	reqd	ATCCIS
requirement	reqt	ATCCIS
reserve	res	ATCCIS
resource	resce	ATCCIS
restriction	restr	ATCCIS
route	rte	ATCCIS
rule-of-engagement	roe	ATCCIS
runway	rway	DLCI
school	schl	ATCCIS
security	secur	ATCCIS
sequence	seq	ATCCIS
serial	ser	ATCCIS
server	svr	ATCCIS
service	svc	ATCCIS
service-feature	sfeat	DLCI
severity	sever	ATCCIS
sheet	sht	ATCCIS
short	short	ATCCIS
size	size	ATCCIS
source	srce	ATCCIS

Annexe A – Convention des Noms

Nom Logique	Nom Physique	Source
spacing	spacg	ATCCIS
speciality	specty	ATCCIS
speed	spd	ATCCIS
standard	std	ATCCIS
start	start	ATCCIS
state	ste	ATCCIS
status	s	DLCI
stock	stk	ATCCIS
storage	stor	ATCCIS
street	str	ATCCIS
subcategory	sbcats	ATCCIS
subject	su	ATCCIS
subtype	styp	DLCI
support	sp	ATCCIS
surface	surf	ATCCIS
surface-region	sregion	ATCCIS
surveillance	surv	ATCCIS
system-parameter	sys_param	DLCI
system	sys	ATCCIS
takeoff	tkoff	DLCI
target	tgt	ATCCIS
task	task	ATCCIS
temperature	tempe	ATCCIS
threshold	throid	DLCI
temporal	tempo	ATCCIS
temporary	temp	ATCCIS
terminal	term	ATCCIS
terrain	ter	ATCCIS
test	test	ATCCIS
text	text	ATCCIS
time	time	ATCCIS
timing	timg	ATCCIS
total	ttl	ATCCIS
track	trk	ATCCIS
transaction	trn	DLCI
transfer	trfer	DLCI
transmittal	transm	ATCCIS
transition	trs	DLCI
lowest-transition	lowtrs	DLCI
* unit	unit	ATCCIS
unit-of-measure	uom	ATCCIS
* unit-type	unitt	DLCI
upper	up	DLCI
usage	usg	ATCCIS
variation	var	DLCI
* vehicle	veh	ATCCIS
verb	verb	ATCCIS
verification	verif	ATCCIS
vertical	vert	ATCCIS
vicinity	vcity	DLCI
visibility	vis	ATCCIS
volume	vol	ATCCIS
voted	voted	DLCI

*Annexe A – Convention des Noms*

<b>Nom Logique</b>	<b>Nom Physique</b>	<b>Source</b>
warning	warn	DLCI
way	way	ATCCIS
weight	wt	ATCCIS
wheeled	whd	ATCCIS
width	wth	ATCCIS
wind	wind	ATCCIS

\* Indique un nom ressemblant à un paramètre requis par CMS.

## Annexe B – ENTITÉS MILML

### DÉFINITION DES ENTITÉS MILML

Tableau de définition des cinq entités clé et leurs rôles

Entité clé	Définition de l'entité	Catégorie d'information
OBJECT-ITEM <sup>1</sup>	An individually identified object that has military significance. Examples are a specific person, a specific item of materiel, a specific geographic feature, a specific co-ordination measure, or a specific unit.	Contents (Who and What)
OBJECT-TYPE	An individually identified class of objects that has military significance. Examples are a type of person (e.g., by rank), a type of materiel (e.g., self-propelled howitzer), a type of facility (e.g., airfield), a type of feature (e.g., restricted fire area), or a type of organisation (e.g., armoured division).	
CAPABILITY	The potential ability to do work, perform a function or mission, achieve an objective, or provide a service.	
LOCATION	A specification of position and geometry with respect to a specified frame of reference. Examples are point, sequence of points, polygonal line, circle, rectangle, ellipse, fan area, polygonal area, sphere, block of space, and cone. LOCATION specifies both location and dimensionality.	Positioning and Shapes (Where)
ACTION	An activity, or the occurrence of an activity, that may utilise resources and may be focused against an objective. Examples are operation order, operation plan, movement order, movement plan, fire order, fire plan, fire mission, close air support mission, logistics request, event (e.g., incoming unknown aircraft), or incident (e.g., enemy attack).	Dynamics (How)

Tableau B-1 Définition des entités MilML<sup>2</sup>

Tableau de définition des sous OBJET-TYPE

Entité	Définition des entités
ORGANISATION-TYPE	An OBJECT-TYPE that represents administrative or functional structures. It is constituted to accomplish an aim, purpose, or mission. The domain values are: Convoy-type, POST-TYPE, UNIT-TYPE, Not otherwise specified.
MATERIEL-TYPE	An OBJECT-TYPE that represents equipment, apparatus or supplies of military interest without distinction to its application for administrative or combat purposes. The domain values are: CONSUMABLE-MATERIEL-TYPE, EQUIPMENT-TYPE, Not otherwise specified.
PERSON-TYPE	An OBJECT-TYPE that represents human beings about whom information is to be held. Example domain values are: Allied military, Civilian, Paramilitary, Prisoner of war, Refugee.
FACILITY-TYPE	An OBJECT-TYPE that is intended to be built, installed or established to serve some particular purpose and is identified by the service it is intended to provide rather than by its content. Example domain values are: Airfield/airport/airstrip, Barracks, Medical facility, POL point, Wire obstacle, double fence.
FEATURE-TYPE	An OBJECT-TYPE that encompasses meteorological; geographic, and control features of military significance. These are naturally occurring phenomena or administratively specified characteristics. The domain values are: CONTROL-FEATURE-TYPE, GEOGRAPHIC-FEATURE-TYPE, Not otherwise specified.

Tableau B-2 Définition des sous entités OBJECT-TYPE<sup>3</sup>

<sup>1</sup> La convention des lettres majuscules est appliquée pour les entités.

<sup>2</sup> CGI, Military Mark up Language - A proposal for Military Interoperability Language based on XML and the NATO LC2IEM, draft version 1.7, 31 mai 2000, page 13.

Tableau de définition des sous OBJET-ITEM

Entité	Définition des entités
ORGANISATION	An OBJECT-ITEM that is an administrative or functional structure
MATÉRIEL	An OBJECT-ITEM that is equipment, apparatus or supplies without distinction as to its application for administrative or combat purposes (e.g., ships, tanks, self-propelled weapons, aircraft, etc., and related spares, repair parts, and support equipment, but excluding real property, installations, and utilities).
PERSON	An OBJECT-ITEM that is a human being to whom military significance is attached.
FACILITY	.An OBJECT-ITEM that is built, installed, or established to serve some particular purpose and is identified by the service it provides rather than by its content (e.g., a refuelling point, a field hospital, a command post).
FEATURE	An OBJECT-ITEM that encompasses meteorological, geographic, and control features of military significance (e.g., a forest, an area of rain, a river, an area of responsibility).

Tableau B-3 Définition des sous entités OBJECT-ITEM<sup>4</sup>

---

<sup>3</sup> Idem, page 28.

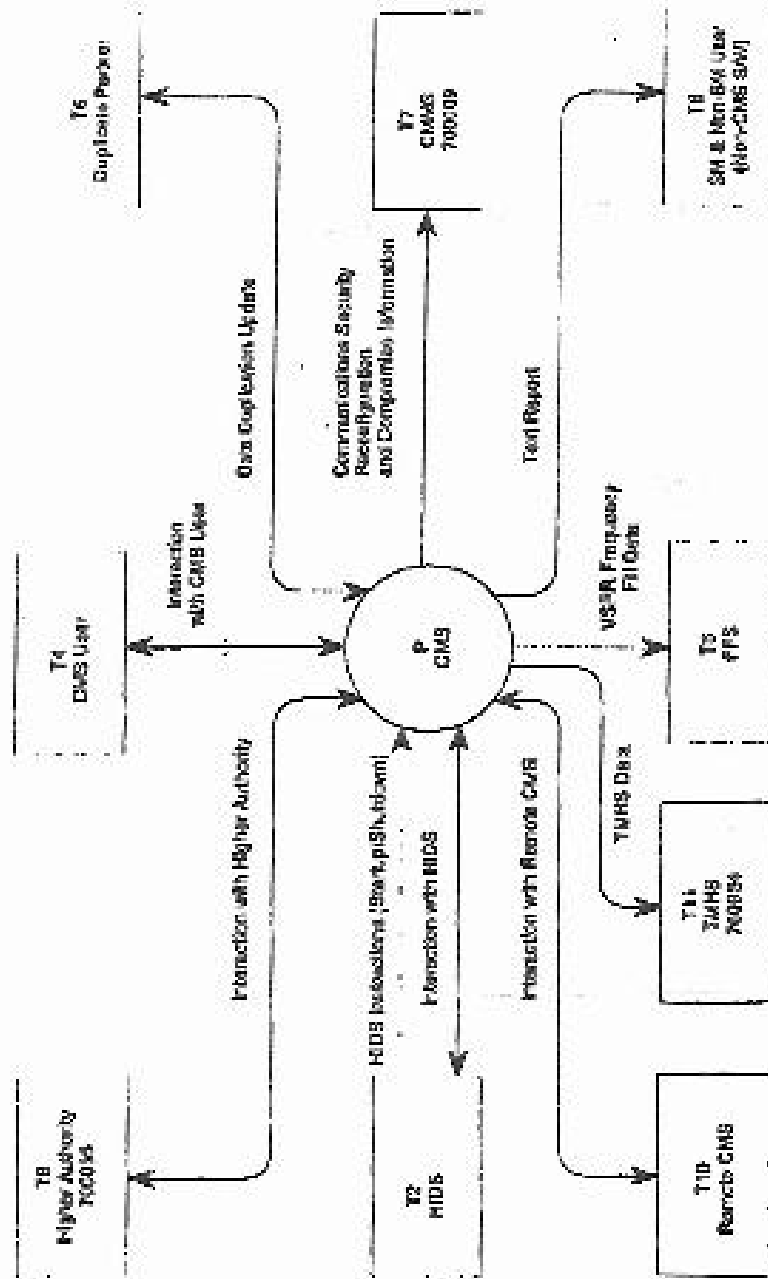
<sup>4</sup> Idem, page 15.



703275K

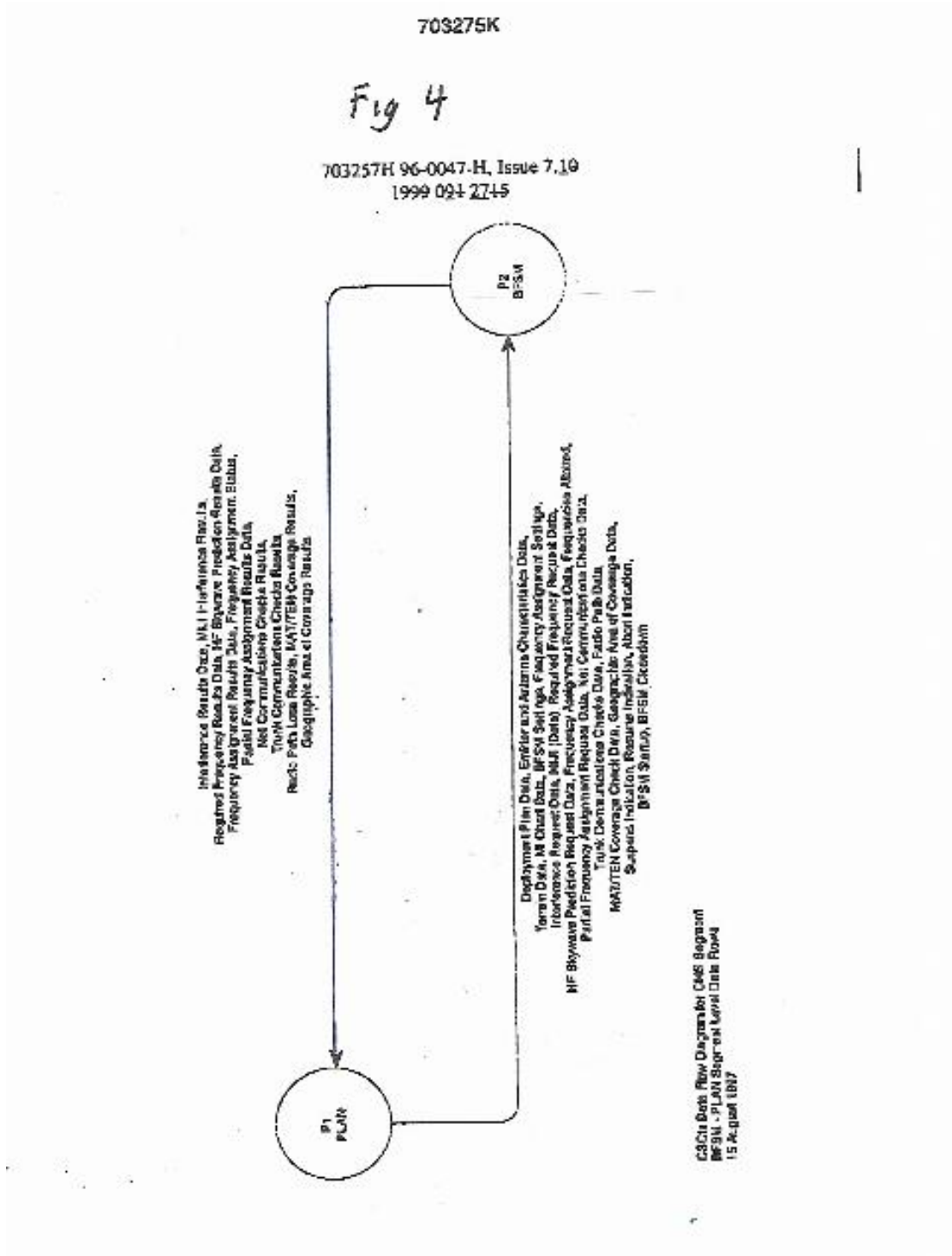
Fig 2

703257H 96-0047-H, Issue 7.10  
1999 09+ 2735



Contract Document for CMS Segment  
15 August 1997





Annexe F

TABLES DE BASE DE DONNÉES PHYSIQUE - OPERA

Référence : LC2IEDM – EBB – Display V3 View – 19 Sep 2001, Data model core based on LC2IEDM 2<sup>nd</sup> Edition (May 2000), exécuté par le Groupe CGI Inc<sup>1</sup>.

Les dix tables de base de données suivantes ont été extraites de la référence pour leur rôle dans le mappage des champs de données de CMS et OPERA. Seul la référence contient le modèle complet.

a. (NATO) UNIT-TYPE - UNITT

	UNITT_ID: numeric(14) (FK)
(ABC)	UNITT_CAT_CODE: varchar(6)
(ABC)	UNITT_MOB_CODE: varchar(6)
	UNITT_PRINC_EQPTT_ID: numeric(14) (FK)
(ABC)	UNITT_SVC_CODE: varchar(6)
(ABC)	UNITT_SIZE_CODE: varchar(6)
(Cdn)	UNITT_APP6A_GRA_PICT_CODE: varchar(6)
(Cdn)	UNITT_LVL_CODE: varchar(6)
	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(14)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
	UNITT_ARM_CODE: varchar(6)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)

b. (Cdn) ORBAT - ORBAT

	ORBAT_ID : numeric(14)
	ORBAT_ORG_ROOT_ID: numeric(14) (FK)
(ABC)	ORBAT_NAME: varchar(50)
(EBB)	ORBAT_DESCRIPTION: varchar(255)
(EBB)	<i>ORBAT_CREATION_DTG: datetime</i>
(EBB)	<i>ORBAT_LAST_MODIF_DTG: datetime</i>
(EBB)	<i>ORBAT_LAST_RECALC_DTG: datetime</i>
(EBB)	<i>ORBAT_PRIVATE_PUBLIC_STATE: varchar(6)</i>
(ABC)	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(14)
(EBB)	LAST_UPDATED: datetime
(EBB)	LAST_UPDATED_BY: varchar(30)
(EBB)	ORBAT_CAT_CODE : varchar(6)

<sup>1</sup> Filière électronique datée du 4 décembre 2001, obtenue de DCIT le 7 décembre 2001.

## c. (NATO) ORGANISATION - ORG

	ORG_ID : numeric(14)
	DISTR_LIST_ID : numeric(14) (FK)
(ABC)	ORG_CAT_CODE : varchar(6)
(ABC)	ORG_NN_NAME : varchar(50)
(Cdn)	ORG_LANG_CODE: varchar(6)
(ABC)	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(14)
(EBB)	LAST_UPDATED: datetime
(EBB)	LAST_UPDATED_BY: varchar(30)
(EBB)	ORG_HIER: varchar(255)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)

## d. (NATO) UNIT - UNIT

	UNIT-ID (FK)
(ABC)	UNIT-FORMAL-ABBREVIATED-NAME
(ABC)	UNIT-IDENTIFICATION-CODE
	UNIT_DISPLAY_ORDER: numeric(6)
	UNIT_IS_HQ_FLAG: varchar(20)
	UNIT_PARENT_AFFILIATION_FLAG : varchar(20)
	UNIT_PARENT_COUNTRY_FLAG : varchar(20)
	UNIT_SPECIAL_SIZE_FLAG: varchar(20)
	UNIT_LAST_UPDATE_DTG: datetime
	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(18)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
	UNIT_INST_QTY: numeric(3)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)

## e. (NATO) ORGANISATION-MATERIEL-ASSOC... OMATA

	OMATA_SU_ORG_ID: numeric(14) (FK)
	OMATA_OB_MAT_ID: numeric(14) (FK)
	OMATA_IDX: numeric(14)
(ABC)	OMATA_CAT_CODE: varchar(6)
	OWNER_ID: numeric(18)
	UPDATE_SEQNR: numeric(18)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
(EBB)	REFRESH_DATE: datetime

f. (NATO) ORGANISATION-STATUS – ORG\_S

ORG_S_ID: numeric(14) (FK)	
ORG_S_IDX: numeric(14) (FK)	
(ABC)	ORG_S_OPER_S_CODE: varchar(6)
(ABC)	ORG_S_OPER_S_QUALF_CODE: varchar(6)
(ABC)	ORG_S_FIRE_MODE_CODE: varchar(6)
(ABC)	ORG_S_RFT_CODE: varchar(6)
(ABC)	ORG_S_RES_IND_CODE: varchar(6)
(ABC)	ORG_S_USG_S_CODE: varchar(6)
	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(18)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)

g. (NATO) MATERIEL-STATUS – MAT\_S

MAT_S_ID: numeric(14) (FK)	
MAT_S_IDX: numeric(14) (FK)	
(ABC)	MAT_S_OPER_S_CODE: varchar(6)
(ABC)	MAT_S_OPER_S_QUALF_CODE: varchar(6)
(ABC)	MAT_S_OPER_S_MODE_CODE: varchar(6)
(ABC)	MAT_S_RES_IND_CODE: varchar(6)
(ABC)	MAT_S_USG_S_CODE: varchar(6)
	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(18)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)

h. (NATO) MATERIEL - MAT

MAT_ID: numeric(14) (FK)	
(ABC)	MAT_SER_NBR_IDENT_TEXT: varchar(50)
(ABC)	MAT_LOT_IDENT_TEXT: varchar(100)
(EBB)	MAT_ALT_IDENT_TEXT: varchar(255)
	OWNER_ID: numeric(14)
	UPDATE_SEQNR: numeric(14)
(EBB)	LAST_UPDATED: datetime
(EBB)	LAST_UPDATED_BY: varchar(30)
(EBB)	REFRESH_DATE: datetime

i. (NATO) EQUIPMENT-TYPE - EQPTT

	EQPTT_ID: numeric(14) (FK)
(ABC)	EQPTT_CAT_CODE: varchar(6)
(ABC)	EQPTT_SBCAT_CODE: varchar(6)
(ABC)	EQPTT_FIRE_GUID_IND_CODE: varchar(6)
	EQPTT_LDED_WT_QTY: numeric(12,3)
	EQPTT_UNLDED_WT_QTY: numeric(12,3)
	EQPTT_MAX_HGT_DIM : numeric(12,3)
	EQPTT_MAX_LHT_DIM: numeric(12,3)
	EQPTT_MAX_WTH_DIM: numeric(12,3)
(ABC)	EQPTT_MLC_CODE: varchar(6)
(ABC)	EQPTT_MOB_CODE: varchar(6)
(ABC)	EQPTT_ECC_CODE: varchar(6)
(ABC)	EQPTT_EQP_CODE: varchar(6)
(ABC)	EQPTT_EQC_CODE: varchar(6)
	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(14)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
	EQPTT_CARGO_IND_CODE: varchar(6)
	EQPTT_RED_HGT_IND_CODE: varchar(6)
	EQPTT_RAIL_TRANSP_IND_CODE: varchar(6)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)

j. (EBB) EQUIPMENT-TYPE-DETAILS – EQPTT\_DTL

	EQPTT_ID: numeric(14) (FK)
	EQPTT_DTL_IDX: numeric(14)
(ABC)	EQPTT_DTL_ATT_NAME: varchar(25)
(ABC)	EQPTT_DTL_ATT_TXT: varchar(255)
	OWNER_ID: varchar(20)
	UPDATE_SEQNR: numeric(14)
	LAST_UPDATED: datetime
	LAST_UPDATED_BY: varchar(30)
(ABC)	RECORD_STATUS: char(1)
	REFRESH_DATE: datetime
	OWNER_HQ: numeric(5)
(ABC)	PRIVATE_TO: varchar(30)
(ABC)	ELEM_VISIBILITY_CODE: varchar(6)

Appendice 1  
Annexe F

TABLES DE BASE DE DONNÉES LOGIQUE - OPERA

Référence : LC2IEDM – EBB – Display V3 View – 19 Sep 2001, Data model core based on LC2IEDM 2<sup>nd</sup> Edition (May 2000), exécuté par le Groupe CGI Inc<sup>1</sup>.

Les dix tables de données suivantes ont été extraites de la référence pour leur rôle dans le mappage des champs de données de CMS et OPERA. Les tables logiques correspondent aux tables physiques du modèle de l'annexe F.

a. (NATO) UNIT-TYPE

unit-type-id (FK)	
(ABC)	unit-type-category-code
(ABC)	unit-type-mobility-code
	unit-type-principal-equipment-type-id (FK)
(ABC)	unit-type-service-code
(ABC)	unit-type-size-code
(Cdn)	unit-type-app6a-graphic-picture-code
(Cdn)	unit-type-level-code
	refresh_date
	owner_hq

b. (Cdn) ORBAT

orbat-id	
	orbat-organisation-root-id (FK)
(ABC)	orbat-name
(EBB)	orbat-description
(EBB)	<i>ORBAT_CREATION_DTG</i>
(EBB)	<i>ORBAT_LAST_MODIF_DTG</i>
(EBB)	<i>ORBAT_LAST_RECALC_DTG</i>
(EBB)	<i>ORBAT_PRIVATE_PUBLIC_STATE</i>
(ABC)	owner_id
	update_seqnr
(EBB)	LAST_UPDATED
(EBB)	LAST_UPDATED_BY
(EBB)	ORBAT_CAT_CODE

<sup>1</sup> Filière électronique datée du 4 décembre 2001, obtenue de DCIT le 7 décembre 2001.



c. (NATO) ORGANISATION

	organisation-id (FK)
	distribution-list-id (FK)
(ABC)	organisation-category-code
(ABC)	organisation-nickname-name
(Cdn)	organisation-language-code
(ABC)	owner_id
	update_seqnr
(EBB)	LAST_UPDATED
(EBB)	LAST_UPDATED_BY
(EBB)	ORG_HIER
	refresh_date
	owner_hq

d. (NATO) UNIT

	unit-id (FK)
(ABC)	unit-formal-abbreviated-name
(ABC)	unit-identification-code
	refresh_date
	owner_hq

e. (NATO) ORGANISATION-MATERIEL-ASSOCIATION

	organisation-materiel-association-subject-organisation-id (FK)
	organisation-materiel-association-object-materiel-id (FK)
	organisation-materiel-association-index
(ABC)	organisation-materiel-association-category-code
(EBB)	last-updated/10
(EBB)	last-updated-by/9
(EBB)	refresh-date/10

f. (NATO) ORGANISATION-STATUS

	organisation-status-id (FK)
	organisation-status-index (FK)
(ABC)	organisation-status-operational-status-code
(ABC)	organisation-status-operational-status-qualifier-code
(ABC)	organisation-status-fire-mode-code
(ABC)	organisation-status-reinforcement-code
(ABC)	organisation-status-reserve-code
(ABC)	organisation-status-usage-status-code
	refresh_date
	owner_hq

g. (NATO) MATERIEL-STATUS

materiel-status-id (FK)	
materiel-status-index (FK)	
(ABC)	materiel-status-operational-status-code
(ABC)	materiel-status-operational-status-qualifier-code
(ABC)	materiel-status-operational-status-mode-code
(ABC)	materiel-status-reserve-indicator-code
(ABC)	materiel-status-usage-status-code
	refresh_date
	owner_hq

h. (NATO) MATERIEL

materiel-id (FK)	
(ABC)	materiel-serial-number-identification-text
(ABC)	materiel-lot-identification-text
(EBB)	materiel-alternate-identification-text
(EBB)	last-updated/13
(EBB)	last-updated-by/13
(EBB)	refresh-date/13

i. (NATO) EQUIPMENT-TYPE

equipment-type-id (FK)	
(ABC)	equipment-type-category-code
(ABC)	equipment-type-subcategory-code
(ABC)	equipment-type-fire-guidance-indicator-code
	equipment-type-loaded-weight-quantity
	equipment-type-unloaded-weight-quantity
	equipment-type-maximum-height-dimension
	equipment-type- maximum-lenght-dimension
	equipment-type- maximum-width-dimension
(ABC)	equipment-type-military-load-classification-code
(ABC)	equipement-type-mobility-code
(ABC)	EQPTT_ECC_CODE
(ABC)	EQPTT_EQP_CODE
(ABC)	EQPTT_EQC_CODE
	refresh_date
	owner_hq

j. (EBB) EQUIPMENT-TYPE-DETAILS

equipment-type-id (FK)	
equipment-type-detail-index	
(ABC)	equipment-type-detail-attribute-name
(ABC)	equipment-type-detail-attribute-text
(ABC)	record_status
	refresh_date
	owner_hq
(ABC)	private_to
(ABC)	elem_visibility_code

Annexe G

TABLES DE BASE DE DONNÉES PHYSIQUE – CMS

Référence : Filière électronique CMS\_DATABASE\_FULL.SQL créé le 21 avril 1999<sup>1</sup>.

Les onze tables de base de données suivantes ont été extraites de la référence pour leur rôle dans le mappage des champs de données de CMS et OPERA. Seul la référence contient le modèle complet.

a. (CMS) DEPLOYMENT\_PLAN

	PLAN_KEY: NUMBER(6)
	SM_ROLE: VARCHAR2(29)
	PLAN_ID: NUMBER(5)
	CREATION_TIMESTAMP: DATE
	CURRENT_PLAN: NUMBER(1)

b. (CMS) UNIT

	UNIT_KEY: NUMBER(6)
	PLAN_KEY: NUMBER(6)
	ICON_NAME: VARCHAR2(45)
	ICON_TYPE: NUMBER(2)
	UNIT_NAME: VARCHAR2(14)
	PARENT_UNIT: NUMBER(6)

c. (CMS) SITE

	SITE_KEY: NUMBER(6)
	PLAN_KEY: NUMBER(6)
	OWNING_UNIT: NUMBER(6)
	ICON_NAME: VARCHAR2(45)
	ICON_TYPE: NUMBER(2)
	SITE_NAME: VARCHAR2(14)
	SITE_REMARKS: VARCHAR2(250)
	TEMP_USED: NUMBER(1)

<sup>1</sup> Filière électronique obtenue du capitaine Clement CMS LCMM/Engr le 7 décembre 2001.

d. (CMS) NODE

	NODE_KEY: NUMBER(6)
	PLAN_KEY: NUMBER(6)
	NODE: NUMBER(6)
	SITE_KEY: NUMBER(6)
	NODE_TYPE: NUMBER(2)
	LOCATION_LAT : NUMBER(10,7)
	LOCATION_LONG : NUMBER(10,7)
	NEXT_LOCATION_LAT : NUMBER(10,7)
	NEXT_LOCATION_LONG : NUMBER(10,7)
	CRASH_LOCATION_LAT : NUMBER(10,7)
	CRASH_LOCATION_LONG : NUMBER(10,7)
	TEMP_LOCATION_LAT : NUMBER(10,7)
	TEMP_LOCATION_LONG : NUMBER(10,7)
	ARRIVAL_TIME: DATE
	DEPARTURE_TIME: DATE
	RERS_NODE: NUMBER(1)
	NODE_SMROLE: VARCHAR2(29)
	NODE_REMARKS: VARCHAR2(250)
	TEMP_USED: NUMBER(1)

e. (CMS) VEHICLE\_STATUS

	VEHICLE_ID: VARCHAR2(40)
	ICON_NAME: VARCHAR2(45)
	ICON_TYPE: NUMBER(2)
	REPORTING_SMROLE: VARCHAR2(29)
	TIMESTAMP: DATE
	VEHICLE_CALLSIGN: VARCHAR2(4)
	VEHICLE_TYPE: VARCHAR2(20)
	SITE_NAME: VARCHAR2(14)
	UNIT: VARCHAR2(14)
	LOCATION_LAT : NUMBER(10,7)
	LOCATION_LONG : NUMBER(10,7)
	VEHICLE_STATE : NUMBER(1)
	TIME_DEPLOYED: DATE
	TIME_RETURNED: DATE
	VEHICLE_REMARKS: VARCHAR2(250)
	OPERATIONAL_ASSIGNMENT: VARCHAR2(200)
	EFD: NUMBER(1)

f. (CMS) VEHICLE

	VEHICLE_KEY: NUMBER(6)
	UNIT_KEY: NUMBER(6)
	SITE_KEY: NUMBER(6)
	ICON_NAME: VARCHAR2(45)
	ICON_TYPE: NUMBER(2)
	VEHICLE_TYPE_NAME: VARCHAR2(20)
	VEHICLE_ID: VARCHAR2(40)
	NODE_KEY: NUMBER(6)
	VEHICLE_CALLSIGN: VARCHAR2(4)
	TEMP_USED: NUMBER(1)

g. (CMS) VEH\_EMIT\_TYPE

	VEHICLE_TYPE_NAME: VARCHAR2(20)
	EMITTER_TYPE_NO: VARCHAR2(2)
	COSITE_FILTER_TYPE: VARCHAR2(20)
	ANTENNA_TYPE_NO: NUMBER(2)
	EMITTER_TYPE: VARCHAR2(20)
	MULTICOUPLER_TYPE_NO: NUMBER(2)
	MULTICOUPLER_PORT: NUMBER(1)

h. (CMS) EMITTER

	VEHICLE_KEY: NUMBER(6)
	EMITTER_NO: CHAR(2)
	PARED_VEHICLE_KEY: NUMBER(6)
	PARED_EMITTER_NO: CHAR(2)
	ANTENNA_NO: NUMBER(2)
	MULTICOUPLER_TYPE_NO: NUMBER(2)
	EMITTER_TYPE: VARCHAR2(20)
	ANTENNA_TYPE_NO: NUMBER(2)
	MULTICOUPLER_PORT: NUMBER(1)

i. (CMS) EMITTER\_TYPE

	EMITTER_TYPE: VARCHAR2(20)
	NOMENCLATURE: VARCHAR2(20)
	CHANNEL_BANDWIDTH: NUMBER(5,1)

j. (CMS) ANTENNA

VEHICLE_KEY: NUMBER(6)
ANTENNA_NO: NUMBER(2)
ANTENNA_TYPE: VARCHAR2(20)
PLANNED_LOCATION_LAT : NUMBER(10,7)
PLANNED_LOCATION_LONG : NUMBER(10,7)

k. (CMS) ANTENNA\_TYPE

ANTENNA_TYPE: VARCHAR2(20)
ICON_NAME: VARCHAR2(45)
ICON_TYPE: NUMBER(2)
ANT_STANDARD_HEIGHT: NUMBER(2)

**Annexe H – XML Application for CMS/OPERA Compatibility**

**XML Application for CMS/Opera Compatibility**  
Delivered to The Department Of National Defense  
On June 6, 2002

**Prepared by:**

**Jeff Kolesnikowicz**

**Code Poets**

10-303 17 Van Order Drive  
Kingston, Ont.



## **Table Of Contents**

Introduction.....	3
XML.....	3
Language Selection.....	3
Program Design .....	6
System Incompatibilities.....	8
CMS Development.....	8
Opera Development .....	11
Incompatible Databases .....	16
PHP Solution VS CGI Solution .....	17

## **Introduction**

The purpose of this project was to provide a demonstration of how XML can allow the transfer of data between two separate, incompatible systems. The systems that were chosen in this proof of concept were the CMS and OPERA systems used by the Canadian Armed Forces.

This document is a history of attempts at developing a solution for this project.

## **XML**

One method to attempt this problem is to develop an XML solution that will import information from the OPERA system to the other system. The following details the steps that were undertaken.

XML is a new language that is intended for transferring data between systems. It can be used to transfer information across networks, to store information in and to format data for other uses. XML is an ideal framework for data exchange.

XML was chosen for this project for a number of reasons.

- XML is an “open” system. Consequently, any developer can produce documentation support for XML and make this documentation publicly accessible thus making it an extremely well supported format.
- XML is written in plain text so it can be moved between systems without the need to convert it from one format to another.
- XML is license free, platform independent and well supported.
- XML is designed for structuring data. It makes it easy for a computer to read, generate and structure the data.
- Humans as well as computers can read XML.
- Another option is a proprietary system. Unfortunately, proprietary systems are not open and as a direct result are not well supported and often costly to initially develop and maintain.
- A ‘data pump’ was being written by CGI that would generate an XML file from the existing Armed Forces OPERA system. This XML file could be transferred to any system.

## **Language Selection**

The next step was to look at what tools are available for a project with these specifications. The project called for the data in an XML document to be imported into the CMS database, so it was important to select the appropriate tool. It was proposed from the start of the project that a custom application would need to be written to import the XML data into the database. The XML was generated from an Opera server by the CGI data pump that was in development.

The criteria for selecting the language were as follows:

- The language must have the facilities to interpret an XML document. This language feature is called an XML parser.
- The program needs to be portable. That is to say, that it can be run on a Windows or a UNIX platform.
- It is necessary for the program to interface with the existing Oracle 7 CMS database.
- It is important to note that the required speed of the application did not influence the choice of language.

There are a number of languages that satisfy the above requirements. Hence, there were many options available. The wide variety of choices was narrowed to C/C++, Java, Perl and PHP, which are described in the following.

C/C++ is an extremely powerful language which is typically used for critical applications. C/C++ is very fast and flexible. Furthermore, it has excellent support since it is one of the most popular languages for programmers to use. On the other hand, C/C++ code need to be *compiled* each time the code is transferred to a new platform. When a program is compiled, it is translated to assembler code so that the computer can understand it. One of the reasons that C/C++ runs so quickly is because it is compiled before it is used. When a program is compiled it is changed to the machine code (the code that the computer requires) for the particular operating system it is running on. Hence, if a program is compiled for Windows, it is not possible to take the compiled program and move it to a UNIX platform. In addition, sometimes the code is not compatible between the platforms. A function call in UNIX might not work in Windows, which means it cannot compile in Windows. To make things worse, many UNIX platforms are not compatible with one other therefore a compiled program cannot be transferred from one version of UNIX to another. It is often the case that the program not only has to be recompiled; it also has to be rewritten for the new platform.

C/C++ is a very fast language once it is compiled properly, but for this particular application speed is not an issue, as the XML would only be imported from time to time. Speed is not an issue in this situation. Moreover, there is an apparent trade off between fast execution time and development time. Execution time is the time a program takes to perform a task. Development time is the amount of time it takes to write an application. It is the case that applications written in Perl or PHP can be written considerably faster than the corresponding<sup>1</sup> application in C/C++. Thus, if execution time is not an issue, then it is unnecessary to spend extra time developing an application in C/C++.

Java is a second programming language choice. Java was developed by Sun Microsystems in the mid-1990's to be a programming language that would work on any platform. The purpose is to circumvent the need to compile it for different platforms. Java's tagline is 'write once, run anywhere'. It accomplished this by forcing the user to

---

<sup>1</sup> c VS Perl: [http://perl.apache.org/perl\\_myth.html](http://perl.apache.org/perl_myth.html)

install a Java interpreter on any machine that was going to run Java programs. An interpreter converts a language into the native machine code for that particular platform. Java is similar in this regard to a scripting language.

A program written in a scripting language is called a script. A script is not compiled until it is run by the system. The interpreter then takes the script and converts it into machine code. This increases the portability of a script since all that is required is an interpreter (on any platform) and the script will run.

A Java program lies between a script and a compiled program since a Java program is compiled and also uses an interpreter. Hence, a Java program can be run on any platform that has the Java runtime environment (an interpreter) available on it. It should be noted that interpreting a language requires an extra step therefore, Java is much slower than C/C++, but again, this is not an important requirement in this application.

Java has many advantages that would facilitate the development of this application. Unfortunately, Java cannot be run on the Armed Forces CMS system. There were two major stumbling blocks, which precluded a Java solution.

1. One problem with running Java with XML and an Oracle database connection is that it needs the proper drivers and modules installed to facilitate the communication of the three tools.
2. A second problem is finding a version of Java that is compatible with the version of UNIX that is running on the CMS machine.

The version of UNIX on the CMS machine is SCO UNIX version 5.0.2. This particular version of UNIX is unsupported at this point in time and has not been updated since 1996. XML is a recent language, XML was not a standard until 1998<sup>2</sup>, hence there are no drivers or modules available for XML with SCO UNIX. Furthermore, Java was not released until 1996; any official version of UNIX that would be found would not have the more modern tools available to it. The reason that Java is unavailable is that, unlike XML, it is a proprietary language, or a “closed” standard. With closed systems only the people with the specifications or source code can produce newer versions of the program. Only Sun Microsystems have the specifications for Java and since SCO UNIX is such an old operating system, they have decided not to produce any new versions of Java for it. Since SCO has an older version of Java and there are no newer versions of Java available for the platform, it would be a time consuming and costly task (if not entirely impossible) to find a compatible XML parser and database driver.

Scripting languages were then examined due to the apparent difficulty of installing Java in SCO UNIX and the lengthy development time necessary to C/C++<sup>3</sup>. Since this is a UNIX environment, Perl and PHP are obvious choices.

---

<sup>2</sup> XML Information: <http://www.w3.org/XML/>

<sup>3</sup> Scripting Language Comparison: <http://www-106.ibm.com/developerworks/library/script-survey/>

Perl is an old language initially developed as a text manipulator. Its first version was released in 1987<sup>4</sup> and the most recent major revision, revision 5, was released in 1995. It has a mature code base and is used by over 1, 000, 000 programmers. Since it has so many uses, Perl has been called “the glue that holds the Internet together”. Perl exhibits many of the same inherent difficulties as Java. Namely, the database and XML modules are not part of the basic installation.

PHP is another scripting language. PHP is newer than Perl, released in 1995, however it is widely supported. It is also an open source language which means that the source code is distributed freely. This is very important to this project since nearly anything that is installed on the CMS machine will need to be recompiled from source code in order to make sure it is compatible. Since it is possible to get the source code, this makes PHP more portable than Java. As long as there is a compiler on the machine, PHP scripts can be run anywhere.

PHP has an XML parser built in thus alleviating the difficulties apparent with Java and Perl. Furthermore, PHP has a built in XML parser and Oracle 7 support. Since it is a scripting language (and does not require compilation), PHP is extremely portable. It is apparent that PHP is the best language to use for this project, since it covers all of the requirements.

## Program Design

The development for this application would be very simple and would depend on the structure of the XML from the OPERA data pump. The most difficult and time-consuming part of the software development is determining the mappings between the XML document and the CMS database.

The input XML would look like this:

```
<items>
  <item type="type_of_item">
    <attribute type="attribute_type" value="value">
  </item>
</items>
```

The items node would have as many items as there were moving from one database to the other. Each item would have as many attributes as there were associated with it. Each item represents an object that is being moved from the OPERA database to the CMS one. An item can be anything that is in the CMS database such as a vehicle, an organization or a site. Inside each item is an attribute. This shows a particular piece of information about a particular item. For example, it might show the VEHICLE\_ID for a vehicle, which would identify a certain vehicle. In addition there would also be a NEXT\_LOCATION\_LAT and a NEXT\_LOCATION\_LONG attribute, which would indicate the next location of the vehicle.

---

<sup>4</sup> Perl Facts: [http://www.perl.org/press/fast\\_facts.html](http://www.perl.org/press/fast_facts.html)

The pseudo-code for the PHP application would look like this:

```
open mappings file, read into AN XML DOM
open XML file, read into an XML DOM
while there are nodes to be read
  read first node
  while there are attributes in node
    search mapping DOM for originating field and table
    combination
    if mapping exists
      check if that already exists in the database
      if it exists
        insert into appropriate directory space
      else
        update the appropriate record
    loop to next attribute
  loop to next node
```

As mentioned previously, the most difficult part of this program is discovering the mappings between the databases and automating the tasks so that when a node is found the application responds by placing the information in the appropriate place. This is a straightforward process. Each field of each table that would need to be mapped would have an individual map node. Inside the node there would be four attributes. The first two would show the originating table and field. The last two would show the fields that the originating information maps to in the CMS database. The XML document would act as a database that uses the attribute type (`o_item_type`) and attribute type (`o_attr_type`) as keys. When the appropriate key has been found, the parser gets the CMS table (`c_table_name`) and field (`c_field_name`) names out next.

With this information, an SQL statement can be made to look up the data from the CMS database.

The mappings file would be an XML document with this form:

```
<mapping>
  <map o_item_type="type_name" o_attr_type="type_name"
  c_table_name="table_name" c_field_name="field_name">
</mapping>
```

There would be more than one map node. Each node would represent a mapping from one database to the other. It is important to note that the user must determine which fields correspond in the two databases since it is impossible for the application to determine this information. The mappings will only need to be set once.

The next step is to build the application.

## System Incompatibilities

A major hurdle that was to be overcome in this project was the incompatibility of the two systems. The systems are incompatible for a number of reasons. Most superficially, they are on two different platforms. One is a Windows NT platform and the other is SCO UNIX. Both operating systems are out of date. The Windows NT platform is old (there have been subsequent releases of Windows 2000 and Windows XP) and is therefore not as supported as a newer operating system is. Consequently, updates are harder to find for this system. Microsoft currently supports Windows NT 4, but it is being phased out.

The SCO UNIX server is unsupported. It had gone through an acquisition by Caldera over a year ago (May 7, 2001) and is no longer supported. When a piece of software is no longer supported, there are no longer any updates being actively developed by the company that owns the software.

The limiting factor with the installation of SCO that is on the CMS system is that it is a custom configuration. There is essentially nothing installed on it in terms of applications. There were no development tools installed on the machine hence no applications can be written on it. A development tool is an interpreter from a programming language to machine code. A programmer will write a program in a particular language and then it is run through a development tool to convert the program to machine code so it can be executed.

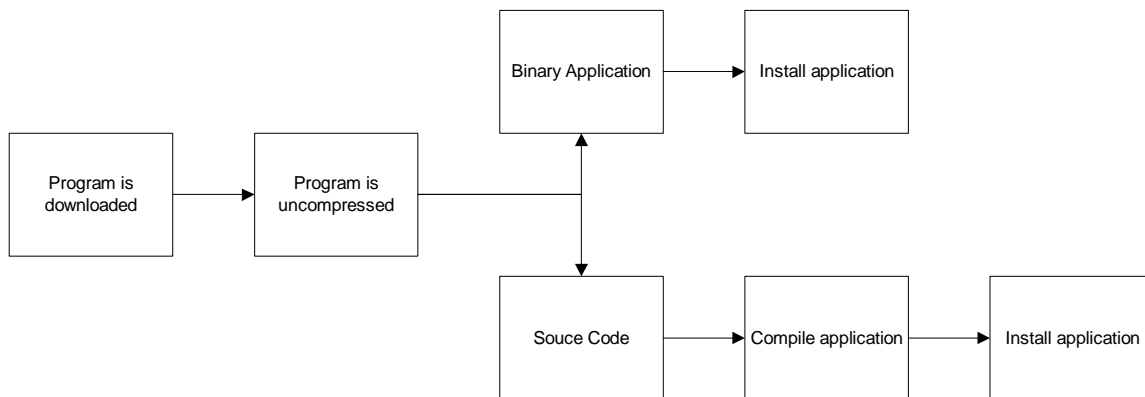
## CMS Development

The CMS system is intended for a use other than development and therefore the designers felt that any development tools were unnecessary. This makes the system fit on very specific hardware. More tools mean that more resources such as storage space and computing power are required to run the application. Since the CMS system was intended to be lightweight and work on minimal hardware, there were the least number of applications installed as possible.

This has proven to be an extremely limiting factor. The SCO installation was pre-configured and there was no utility disk provided. If a copy of the installation disk could have been found, it would have been possible to install some of the tools that were needed.

There are two ways to install new programs on a UNIX platform. The first is to get a binary copy of the application and install it either through an installation script or through the package manager. The SCO package manager is called '*custom*'. When the *custom* command is run, a menu system allows the user to install a new program. The second way to install programs is to acquire the program's source code and then compile it. The Caldera website had a copy of PHP on its download page (<http://www.caldera.com/skunkware/>). The version of PHP was downloaded onto the machine.

The following diagram shows the steps in installing a new package on a UNIX platform.



The first step is to acquire the application that needs to be installed. The next step is to uncompress it. At this stage it can be determined if the application is a binary or simply the source code. If it is a binary application it can be installed using the appropriate installation utility. In the case of the CMS machine the utility is called *custom*. If the application is just the source code then it needs to be compiled and then it can be installed.

In order to install an application on a machine the following utilities are required:

- A decompression utility. On UNIX it depends on what format the package was compressed using. Common formats are: tar, gzip and Z. This was missing from the CMS machine
- An application installer. On the CMS machine this is the *custom* utility. This was installed:
- A compiler. This is for applications that are distributed in source code. A common compiler on the UNIX platform is GCC. This was also missing from the CMS machine.
- Binary utilities, called binutils. These go hand in hand with the compiler. They convert the source code into machine code. This was also missing from the CMS machine.
- In order to develop an application for a UNIX machine, an interpreter is needed. In the case of this project the PHP one is needed. Other interpreters include Java, Perl or C/C++.

Clearly, many of the required pieces of the machine are missing. Since this is a closed system, technical support was required. The project manager for the CMS application was hesitant to help because the project being developed was outside of the scope that he was asked to develop.



The next problem was that the installation package was compressed and there was no uncompression utility installed on the machine. The particular utility that was missing is called gzip. The utility was downloaded it from the gzip page and installed it. The URL is here: <http://www.eso.org/mirror/gzip/gzip-1.2.4.x86-sco-opensv5.0.2.tar.Z>. To install the program you need to run these commands:

```
mkdir gzip
mv gzip* gzip/
uncompress gzip*
tar -xvf gzip*
mv gzip /usr/local/bin/
```

Luckily, the gzip utility was available for the version of SCO UNIX in binary format and didn't need to be compiled.

Next the can uncompress any gzipped file using the command: `gzip -d <filename>` and then installed using the custom command.

There were some problems with the PHP installation. First it was an older version without XML support. That is a very critical problem. Second, there was no support for Oracle on it. In order for PHP to be able to work with Oracle, it needs to be compiled to use the proper commands.

The source code the source code from php.net was downloaded (<http://www.php.net/distributions/php-4.2.0.tar.gz>), uncompressed it by running these commands:

```
gzip -d php*
tar -xvf php*
```

The steps to install a program from source code are as follows:  
(from the directory with the uncompressed code)  
`./configure`  
`make`  
`make install`

In order to get the proper functionality built into PHP, the following options need to be added to the `./configure` command.

- To install the XML parser the command `--with-XML` needs to be added.
- Next, PHP has a built in Oracle library. To set it up, the installed needs to point the compiler at the Oracle database and it will do the rest. This option is configured by entering the `--with-oracle=/path` option in the `./configure` step.

When the '`./configure`' command, the computer gave the error 'gcc not installed'. This means that GCC is not installed. GCC is a compiler for the UNIX platform.

In order to compile anything, the system needs a copy of a c compiler (GCC or CC) and a copy of the binary utilities (binutils). The C compiler allows users to download and install new applications for the system. The binutils contain a copy of the linker and the assembler, which allows for source code to be converted into machine code. These are essential utilities in installing new software from source code, but they were all missing from the CMS machine.

On the Caldera site, there is a copy of the GCC compiler available for download. It was downloaded and installed using the ‘custom’ program. When the ‘./configure’ command was run again, the computer gave an error message back that said ‘*In not found*’. This was the binutils package missing.

On the Caldera website and it said that the binutils weren’t available for download. It is possible to download the source code for binutils, but it isn’t possible to compile that code without the binutils. Binutils was found at a website that was taken down soon after it was downloaded. This version of binutils was precompiled for the SCO UNIX platform. It was installed using the *custom* command.

After binutils was installed, the ./configure command was run again. Still PHP refused to compile because it said the version of GCC and binutils was too old to compile PHP. The version of GCC and binutils also had problem compiling a newer version of GCC. As it stands now, the version of GCC is not acceptable for compiling a modern version of PHP and will need to be upgraded.

## Opera Development

The Opera machine was setup according to the requirements in the Opera installation guide. The Opera software installed without a problem.

The data pump had some problems being installed. The instructions were followed from the “XML Generated Detailed User Guide”, but the following error was generated:

```
..\jre\bin\java -server -Xms8m -Xmx128m -showversion -DDebug=true
-Duser.dir=e:\ods -Dfile.encoding=ISO-8859-1 -classpath ./lib/utility.jar;./lib/xml4j.jar;./lib/classes12.jar;./lib/jce1_2_1.jar;./lib/sunjce_provider.jar;./lib/DataPump.jar -DLoggingLevel=VERBOSE ca.dnd.land.itl.pump.Launcher
\xml\DIHRS_
2001\SECT_TYPE\DIHRSSectType.xml
java version "1.3.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.0-C)
Java HotSpot(TM) Server VM (build 2.0fcs-E, mixed mode)

Exception in thread "main" java.lang.NoClassDefFoundError
```

CGI was called for technical support and they said this was a known error. They also said that there was a new solution that had been developed.

Since this data pump did not work, a new solution would have to be developed to extract XML from the Opera side.

### **Description of CGI Solution**

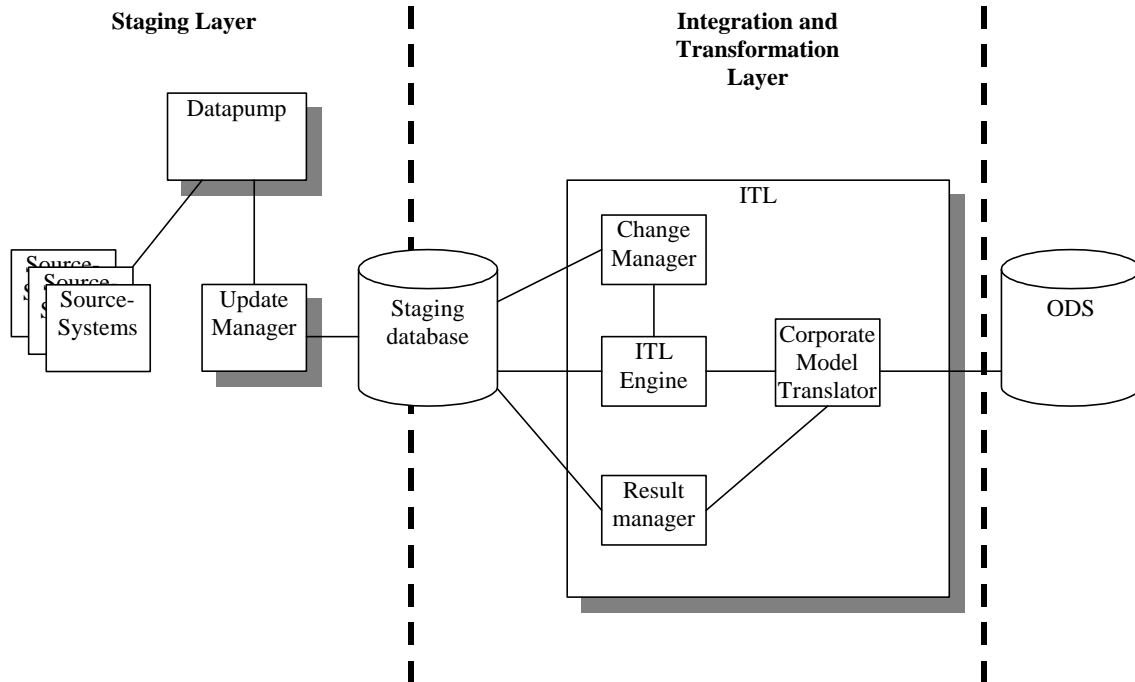
Another solution that was offered by CGI was a program called Generic Enterprise Data Integration System (GEDIS). This is a Java application that was developed by CGI in order to allow users to store their data into a common Operational Data Store (ODS). This intent of this application is to be able to use it on any platform using any relational database for the ODS. GEDIS is intended to be an enterprise-sized solution.

According to Maxime Tardif, a CGI consultant who worked on the GEDIS project, the requirements for this application are as follows:

- JRE 1.3 (Java).
- A relational database (Oracle, SQL Server...) as staging database.
- Appropriate JDBC drivers (for remote systems and staging database).
- xerces.jar for parsing XML

The following is taken from the General Architecture document for the GEDIS system.

This is a high level view of GEDIS:



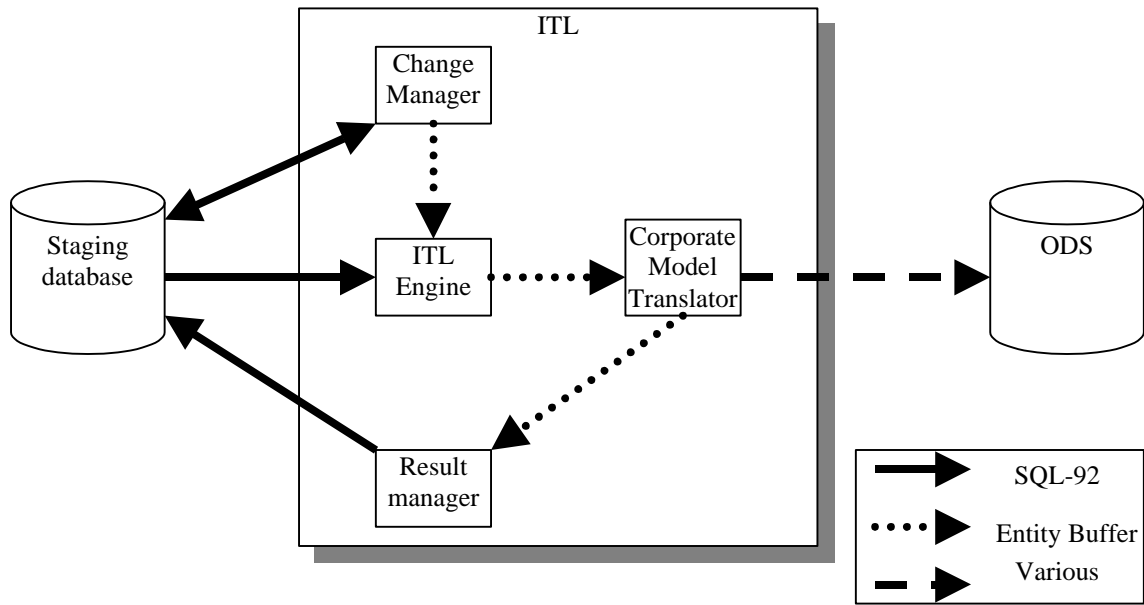
This is a description of each component as taken from the General Architecture document provided for the GEDIS system. Each component of the above diagram has a short description below:

Component	Sub-Components	Implemented	Roles and responsibilities
Datapump	-	Yes	The datapump is responsible for extracting source-system information in various sources and transforming it into XML documents.
Update manager	-	Yes	The update manager is responsible for keeping a local copy of source system data in the staging database. It is also responsible for determining which one has change since the last verification. It changes the data status in the staging database.
Staging database	-	Yes, schemas	The staging database contains the last image of the legacy system that is waiting to be processed. Also, the staging database contains information for integration, transformation and configuration of the system.

ITL	Change manager	Yes	Change manager is responsible for taking the marked data to be send to the ODS. Also, it is responsible for insuring the synchronization between update manager and ITL Engine.
	ITL Engine	Yes, may need extensions for specific processing contexts	Responsible for transforming and integrating data
	Corporate model translator (CMT)	Yes, need to be extended to specific ODS database technologies	Responsible for pushing the data into the ODS. The general frame of CMT is developed; it needs to be extended to specific ODS technologies. (SQL, CORBA, XML, SOAP, DCOM ...)
	Result manager	Yes	Responsible for logging the execution status of the ITL process. That component is located at the end of the ITL layer process and perform the synchronization of the data after they have been touched by the CTM. It take data in Entities object adjust the corresponding row in the staging database.
ODS	-	No	Operational Data Store database, used to keep integrated and transformed data on Corporate Model for analytic applications.

The source systems can be nearly any type of file. Since the GEDIS system can uses JDBC drivers (ODBC drivers for Java), nearly any driver can be plugged in and run. The source can be any number of sources including XML or flat files. The data that is processed is then translated using the Corporate Model Translator and can make the data into any format type that is installed on the computer. In the case of the project at hand, the data would be exported from the

This is a closer look at the translation step:



Each component is described below:

From	To	Bi-directional	Type	Content
Stating database	Change Manager	Yes	SQL-92	Data in the staging database are marked as “in process” and then are put in memory.
Change Manager	ITL Engine	No	Entity Buffer	The Change Manager puts newly retrieved entities form the database into the Entity Buffer for waiting to transformation by the ITL engine.
ITL Engine	Corporate Model Translator	No	Entity Buffer	The ITL Engine puts the transformed entities into the Entity Buffer to waiting for the processing by the corporate model translator.
Corporate Model Translator	Result Manager	No	Entity Buffer	The CMT puts sent entities in the Entity Buffer to be

				synchronised within the staging database by the result manager
Corporate Model Translator	ODS	No	Various	The way that the CMT transfer information to the ODS depends in the specific implementation of the Corporate Model Translator.
Staging database	ITL Engine	No	SQL-92	The ITL Engine takes information about integration and transformation from the staging database.
Result Manager	Staging database	No	SQL-92	The Result Manager puts the entities in staging in their right final status and puts error information in the database if necessary.

### Incompatible Databases

Since the CMS and Opera application were developed separately, the database are incompatible. XML solves the problem of transferring the information in a compatible format because it is in a text format and any operating system can read text. The problem here lies with the naming convention used in the databases.

Two different standards and naming conventions were used between the two databases. For example in the CMS system a vehicle is represented in the VEHICLE table. In the Opera database, a vehicle is represented in the EQPTT table. The differences do not end there. The field VEHICLE\_TYPE\_NAME represents the type of vehicle in the CMS database. In the Opera database, the same field is named EQPTT\_ID. Another example from the same table in CMS is VEHICLE\_ID. This field represents the exact ID of the vehicle. In the Opera database the same field is named EQPTT\_ECC\_CODE.

The naming convention is not the only incompatibility between the databases. The sizes of the fields are also different. For example the VEHICLE\_TYPE\_NAME field in the CMS database is the type VarChar(20). This means the field will accept any character type, but only up to 20 characters. The Opera side is different. EQPT\_ECC\_CODE is a VarChar(6), which means it can only take up to 6 characters. The differences in size indicate that the two different fields hold different data. The Opera database holds a pointer to a different table, which holds the information. A

different table will hold the actual information. This is how a relational database stores the information. The CMS database is not relational for this portion of the design.

### **PHP Solution VS CGI Solution**

The PHP solution offers a different solution than the GEDIS solution offered by CGI.

The PHP solution described here is a custom application. It is written to do one job with a simple interface. Once it is written, it would not need updating or maintenance as long as the OPERA and CMS databases remained the same. This solution would be a very simple approach and would only do the simple task of entering a very specific XML structure into the CMS Oracle database. The program would handle all of the mappings between the OPERA and CMS databases.

The GEDIS solution on the other hand is a more generic structure. It is designed to work on a number of platforms with a number of databases. It is an extremely flexible tool. The tradeoff here is complexity. According to a contact at CGI, Marie Théberge ([marie.theberge@cgi.ca](mailto:marie.theberge@cgi.ca)), it would take a new user two weeks to be trained on the GEDIS system. This is a non-trivial amount of time. The time estimated to complete the PHP solution described above is estimated at 20 hours including testing and deployment.

Despite the complexity of the GEDIS program, it is unable to predict the format of the destination database. The user would be required to take the mappings of the OPERA information and feed them into the CMS scheme. Since this is the biggest part of the project, there is no savings over building a custom application. It would make more sense taking the two weeks of training required for the GEDIS application and developing a custom application.



## **Bibliographie**

Accredited Standard Committee, What is ASC X12, <http://www.x12.org/x12org/about/whatis.html> , 7 mai 2002.

Agnoux, Hervé, Le consortium OASIS a déclaré que ses travaux autour de SAML étaient proches du niveau « maturité », <http://xmlfr.org/actualites/tech/020513-0001>, 13 mai 2002.

Ahmed, Kal et autres, Professional XML Meta Data, (ISBN: 1861004516), publié par Worx Press, 2001

Birbeck, Mark et autres, Professional XML Second Edition, (ISBN: 1861005059), publié par Wrox Press, 2001

Box, Don, "Lessons from the Component Wars: An XML Manifesto", <<http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnxml/html/xmlmanifesto.asp>>, Septembre 1999, Copyright © 2001 Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399 U.S.A. All rights reserved.

BP STCCC/SLI, Aperçu du Système IRIS, C-53-750-000/MA-001, 26 mars 1999, page 4-2-1.

CGI et Centre de Recherche de la Défense de Valcartier, Opera Installation Guide (Draft), version 0.1, 14 avril 2002, page 2.

CGI, Military Mark up Language - A proposal for Military Interoperability Language based on XML and the NATO LC2IEM, draft version 1.7, 31 mai 2000, page 9.

Chase, Nicholas, XML et Java, (ISBN: 2-7440-1166-5), publié par CampusPress, 2001

Chazalon, Grégory, Lemoine, Joséphine, "Les schémas XML", <[http://zuse.esnig.cifom.ch/intranet/SpecSchema/xschema.htm#\\_Toc504748122](http://zuse.esnig.cifom.ch/intranet/SpecSchema/xschema.htm#_Toc504748122)>, 21 janvier 2001.

Chef d'état-major de l'Armée de Terre, Directives stratégiques sur les opérations et les ressources 2002, Chapitre 3, Section 2, Annexe A, page 4.

Chem eStandards™ Frequently Asked Questions, [http://www.cidx.org/cidx/faq/faq\\_02.html](http://www.cidx.org/cidx/faq/faq_02.html), 7 mai 2002

Chem eStandards™ Frequently Asked Questions, [http://www.cidx.org/cidx/faq/faq\\_02.html](http://www.cidx.org/cidx/faq/faq_02.html), 6 mai 2002.

Clement, Capt Jason, BP STCCC, Courriel "FWD SCO Compiler for CMS", 15 mai 2002.

Computer Reseller News, "MOG pushes to become standards body, touts CORBA", <<http://www.techweb.com/se/directlink.cgi?CRN19970728S0027>>, juillet 1997

Cover, Robin, Committee Specification Level Documents for the Security Assertion Markup Language (SMTP), < [http://xml.coverpages.org/ni\\_2002-04-20-a.html](http://xml.coverpages.org/ni_2002-04-20-a.html)> 5 mai 2002, 20 avril 2002.

DCIT, "DLCI Command Support Brief to 2 CMBG", 6 septembre 2001.

Defence Plan 2001, Chapter 5, Article 2.1.2.

DISA/JIEO, "DII COE Distributed Applications Series - Recommendations for Using DCE, DCOM, and CORBA Middleware", 13 avril 1998, Figure 1, page 14

DLIR, Army Integrated Management Environment (AIME) Overall Technical Architecture v 1.7, 12 March 2001.

Duckett, John et autres, Professional XML Schemas, (ISBN: 1-861005-47-4), publié par Worx Press, 2001

ebXML Catalog of Common Business Processes v1.0, 11 May 2001,  
<http://www.ebxml.org/specs/bpPROC.pdf>, page 9.

ebXML, Enabling A Global Electronic Market – General Information  
<http://www.ebxml.org/geninfo.htm>, 6 mai 2002.

Eliotte Rusty Harold, XML Bible second Edition, (ISBN: 0-7645-4760-7), publié par Hungry Minds, 2001

Extrait de "Namespaces in XML", <http://www.w3.org/TR/1999/REC-xml-names-19990114>, Copyright © 1999 W3C (MIT, INRIA, Keio), W3C, W3C RECOMMENDATION, 6 mai 2002.

Giguère, Charles, "Comprendre XML", <<http://membres.lycos.fr/cgiguere/vdn/vdn85.htm>>, 2001.

Groupe CGI, "Data Pump Component Technical Architecture V2.2", 12 juillet 2000.

Groupe CGI, "Integration and Transformation Layer (ITL) Technical Architecture V1.2", 15 juillet 2000.

Groupe CGI, "Synchronization Technical Architecture V1.2", 17 juillet 2000.

Groupe CGI, "Update Manager Technical Architecture V1.3", 28 juillet 2000.

Groupe CGI, "XML Transformer V 1.2", 17 juillet 2000.

Groupe CGI, "Introduction to ATCCIS Information Session presented to Land Forces Staff", Mars 2000

Guerrero Ignacio, "OPENGIS XML- BASED WEB MAP INTERFACES",  
<[http://www.canri.nsw.gov.au/activities/dsd2001/workshop2/wkshp2\\_intergraph.htm](http://www.canri.nsw.gov.au/activities/dsd2001/workshop2/wkshp2_intergraph.htm)>, 13 mai 2002

How CHEM eStandards™ Were Developed, [http://www.cidx.org/cidx/commerce/how\\_dev.html](http://www.cidx.org/cidx/commerce/how_dev.html), 6 mai 02.

HR-XML Consortium, About HR-XML, <http://www.hr-xml.org/channels/about.cfm>, 5 mai 2002.

HR-XML Consortium, SIDES Fact Sheet, <http://www.hr-xml.org/sides/sidesfactsheet.cfm>, 5 mai 2002.

Ibrahim, Bertrand, "Schémas XML", <<http://cuisung.unige.ch/xml/Schemas.html>>, 10 mai 2001.

## Bibliographie

InternetWeek.com, Securing XML At The Data Level,  
<http://www.internetwk.com/story/INW20020503S0002>, 3 mai 2002.

JNet Solutions, Le W3C complète le chapitre de la sécurité des services Web,  
[http://solutions.journaldunet.com/0203/020322\\_xkms.shtml](http://solutions.journaldunet.com/0203/020322_xkms.shtml), 22 mars 2002, 10 mai 2002.

Kindel, Charlie and Williams, Sara, "The Component Object Model: A Technical Overview",  
<[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn\\_comppr.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn_comppr.asp)>,  
Octobre 1994

Kolesnikowicz, Jeff, XML Application for CMS/Opera Compatibility, livré le 6 juin 2002.

Koller, Mike, Pushed Off The Fence, <http://www.internetweek.com/newslead01/lead112601.htm>, 26  
November 2001.

Kotok, Alan, Government and Finance Industry Urge Caution on XML, April 24, 2002.  
[www.xml.com/pub/a/2002/04/24/gaonacha.html](http://www.xml.com/pub/a/2002/04/24/gaonacha.html), 6 mai 2002.

Le Chemical Industry Data eXchange (CIDX™), Un livre blanc sur le CIDX,  
[http://www.cidx.org/cidx/res\\_library/index.html](http://www.cidx.org/cidx/res_library/index.html), 6 mai 2002.

Levesque, Claude, major, Impacts of XML on Canadian Army C2IS, Document militaire d'un officier d'échange SICF/LFCS,  
Paris, 1 juin 2000.

Lindstrom, Pete, Special Report: The Language of XML Security,  
<http://www.networkmagazine.com/article/NMG20010620S0001>, Network Magazine 5 juin 2001

Lindstrom, Pete, Special Report: The Language of XML Security,  
<http://www.networkmagazine.com/article/NMG20010620S0001>, Network Magazine 5 juin 2001

Lt. N. Pedneault & Lt. B. Cornell, 2 CMBG TCCCS Technical Coordination Cell, Communication  
Management Aide-Mémoire, sans date, 49 pages.

Maj JR Couture, Weisman, Bob, "Army Integrated management Environment Phase 2 – Proof of Concept  
Brief", 1<sup>er</sup> mars 2000.

Microsoft Corporation, "DCOM Technical Overview",  
<[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn\\_dcomtec.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomtec.asp)>,  
Novembre 1996

Multilateral Interoperability Programme (MIP), MIP Interface Operating Procedures  
- FR-PWG – Version A3, 21 septembre 2001.

OASIS, About LegalXML, <http://www.legalxml.org/about/index.shtml>, 6 mai 2002.

## Bibliographie

OCTO Technology, Le Livre Blanc de la Sécurité, © 2001 OCTO Technology, septembre 2001, « [www.octo.com/fr/techno/download/wp-secu.pdf](http://www.octo.com/fr/techno/download/wp-secu.pdf) », 5 mai 2002), 96 pages. - Les Livres Blancs d'OCTO Technology sont gratuits et diffusables à volonté dans leur version originale. Tout extrait ou diffusion partielle est interdite sans autorisation préalable d'OCTO Technology -

Pillou, Jean-François, "eXtensible Markup Language", <<http://www.commentcamarche.net/xml/xmlintro.php3>>, 2001

Pillou, Jean-François, "Parser du XML – Les API DOM et SAX", <<http://www.commentcamarche.net/xml/xmldomsax.php3> >, 2001.

PMO TCCCS, Interface Control Document (ICD), 23 August 2001.

Présentation BP STCCC, Briefing to LFTSP, 14 février 2002.

Raj, Geopalan Suresh, "A Detailed Comparaison of CORBA, DCOM and Java/RMI", mai 2002, Copyright © 1997-2001 Gopalan Suresh Raj, 8730, S. Country Drive, Apt #201, Oak Creek, WI 53154-3880 U.S.A. All rights reserved.

RosettaNet Overview, <http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?Container=com.webridge.entity.Entity%5BROID%5B36249C27FC2BD411841F00C04F689339%5D%5D>, 6 May 2002

RosettaNet, Standards, [http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?expanded=com.webridge.entity.Entity\[OID\[5F6606C8AD2BD411841F00C04F689339\]\]&container=com.webridge.entity.Entity\[OID\[5F6606C8AD2BD411841F00C04F689339\]\]](http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?expanded=com.webridge.entity.Entity[OID[5F6606C8AD2BD411841F00C04F689339]]&container=com.webridge.entity.Entity[OID[5F6606C8AD2BD411841F00C04F689339]]), 5 mai 02

Secure rights to digital assets with ODRL, <http://www.odrl.net/1.0/ODRL-10-HTML/ODRL-10.html>, 16 mai 02.

Segue Software, "CORBA Primer", <<http://cgi.omg.org/library/seguecorba.pdf>>, 2002

soapuser.com, "Qu'est-ce que SOAP?", <<http://www.soapuser.com/fr/basics1.html> >, 2001.

Sun Microsystems, "An Overview of RMI Applications", <<http://java.sun.com/docs/books/tutorial/rmi/overview.html>>, 18 mai 2002, Copyright 1994-2002 Sun Microsystems, Inc.

Sun Microsystems, "Java™ Remote Method Invocation – Pure Java Distributed Computing", <[http://java.sun.com/marketing/collateral/rmi\\_ds.html](http://java.sun.com/marketing/collateral/rmi_ds.html)>, 18 mai 2002, Copyright 1994-2002 Sun Microsystems, Inc.

Sun Microsystems, "RMI System Overview - Chapter 3.5: RMI Through Firewalls Via Proxies", section 3.5.5, <<http://java.sun.com/j2se/1.4/docs/guide/rmi/spec/rmi-arch6.html>>, 18 mai 2002, Copyright 1994-2002 Sun Microsystems, Inc.

The Goals of MathML, <http://www.w3.org/Math/mathml-faq.html#goals>, 6 mai 2002.

Transentric, A White Paper, TranXML™ : The Common Vocabulary for Transportation Data Exchange, [http:// www.transentric.com/products/commerce/transWP.pdf](http://www.transentric.com/products/commerce/transWP.pdf), 5 mai 2002, page 1.

W3C Recommendation 21 February 2001, [www.w3.org/TR/2001/REC-MathML2-20010221](http://www.w3.org/TR/2001/REC-MathML2-20010221), 6 mai 2002.

Watts, John, DLCSPM 2-4, SCO Compiler for CMS, 14 May 2002.

Web and Web services Magazine, SAML Advances Single Sign-On Prospects, [http://www.fawcette.com/xmlmag/2002\\_03/magazine/departments/marketscan/SAML/](http://www.fawcette.com/xmlmag/2002_03/magazine/departments/marketscan/SAML/), 13 mai 2002.

Weisman, Robert, Presentation on AIME – Environment Phase 2 Proof of Concept, Groupe CGI Inc, 16 March 2000.

XBRL, Overview Fact Sheet, <http://www.xbrl.org/Overview.htm> , 5 mai 02.

Conformité aux termes d'utilisation des documents disponibles sur le site web de MSDN pour l'utilisation de l'article de Don Box: "Lessons from the Component Wars: An XML Manifesto", <<http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnxml/html/xmlmanifesto.asp>>, Septembre 1999

**NOTICE SPECIFIC TO DOCUMENTS AVAILABLE ON THIS WEB SITE.**

Permission to use Documents (such as white papers, press releases, datasheets and FAQs) from the Services is granted, provided that (1) the below copyright notice appears in all copies and that both the copyright notice and this permission notice appear, (2) use of such Documents from the Services is for informational and non-commercial or personal use only and will not be copied or posted on any network computer or broadcast in any media, and (3) no modifications of any Documents are made. Accredited educational institutions, such as K-12, universities, private/public colleges, and state community colleges, may download and reproduce the Documents for distribution in the classroom. Distribution outside the classroom requires express written permission. Use for any other purpose is expressly prohibited by law, and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible.

**COPYRIGHT NOTICE.** Copyright © 2001 Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399 U.S.A. All rights reserved.

Conformité aux termes d'utilisation des documents disponibles sur le site web de Sun Microsystems pour l'utilisation de l'article Sun Microsystems, "RMI System Overview - Chapter 3.5: RMI Through Firewalls Via Proxies", section 3.5.5, <<http://java.sun.com/j2se/1.4/docs/guide/rmi/spec/rmi-arch6.html>>, 18 mai 2002, , Copyright 1994-2002 Sun Microsystems, Inc.

### **13. Intellectual Property Rights**

13.1 The Services and any software used in connection with the Services ("Software") contain proprietary and confidential information that is protected by applicable intellectual property and other laws. Content contained in third-party advertisements or information presented to You through the Services or advertisers is protected by copyrights, trademarks, service marks, patents, publicity rights, or other proprietary rights and laws. Except as expressly authorized by Sun or advertisers, You agree not to modify, rent, lease, loan, sell, distribute or create derivative works based on the Services or the Software, in whole or in part.

13.2 You may use the Software and related Services only subject to the agreement or license that accompanies such Software or Services. You may use Services only during their specified term, for the enumerated number of times, for the specified number of individuals, or as otherwise limited in the applicable agreement or license. Services involving an update, version release, product release, maintenance release, patch or derivative work of a licensed Software or Service may be used only: (a) on systems for which such Software or Service were specifically licensed; and (b) subject to the license and warranty term governing the original product or Service. Services provided as an element of Sun's support, consulting or educational services, also are governed by the terms of the applicable support, consulting or educational services agreement.

13.3 You must not modify, decompile, or reverse engineer any Software Sun discloses to You, and must not remove, overprint, or deface any notice of copyright, trademark, logo, legend, or other notice of ownership from any originals or copies of Software or Information from the Services.

13.4 "Sun Trademarks" means all names, marks, brands, logos, designs, trade dress and other designations Sun uses in connection with Products or Services. Customer may refer to Products and Services by the associated Sun Trademarks, provided that such reference is truthful and not misleading and complies with the then current Sun Trademark and Logo Policies, which are located at <http://www.sun.com/policies/trademarks>. Customer may not remove or alter any Sun Trademarks, nor may it co-logo Products or material associated with Customer's services. Customer acknowledges Sun's rights in Sun Trademarks and agrees that any and all use of Sun Trademarks by Customer shall inure to the sole benefit of Sun. Customer agrees not to incorporate any Sun Trademarks into Customer's trademarks, service marks, company names, Internet addresses, domain names, or any other similar designations.

13.5 Sun is committed to respecting others' intellectual property rights, and we ask our users to do the same. Sun may, in its sole discretion, terminate the accounts or access rights of users who infringe or otherwise violate others' intellectual property rights. If You believe that Your work has been copied in a way that constitutes copyright

infringement on our Website, please contact our copyright agent as described in our [copyright policy](#).

13.6 All Content provided on this Website is provided by or to Sun by its respective manufacturers, authors, developers and vendors (the "Third Party Providers") and is the copyrighted work of Sun and/or the Third Party Providers. Except as stated herein, none of the Content may be copied, reproduced, distributed, republished, downloaded, displayed, posted or transmitted in any form or by any means, including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior express written permission of Sun or the Third Party Provider. No part of the Website, including logos, graphics, sounds or images, may be reproduced or retransmitted in any way, or by any means, without the prior express written permission of Sun. You also may not, without Sun's prior express written permission, "mirror" any Content contained on this Website on any other server.

13.7 Nothing on this Website shall be construed as conferring any license under any of Sun's or any Third Party Provider's intellectual property rights, whether by estoppel, implication, or otherwise. You acknowledge sole responsibility for obtaining any such licenses.

13.8 Permission is granted to display, copy, distribute and download Sun's Content on this Website provided that: (1) both the copyright notice identified below and this permission notice appear in the Content, (2) the use of such Content is solely for personal, non-commercial and informational use and will not be copied or posted on any networked computer or broadcast in any media, except as explicitly permitted by valid license covering such materials, and (3) no modifications of any of the Content are made. This permission terminates automatically without notice if You breach any of these terms or conditions. Upon termination, You must immediately destroy any downloaded and printed Content.

13.9 Any unauthorized use of any Content contained on this Website may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

13.10 Content provided by Third Party Providers has not been independently authenticated in whole or in part by Sun. Sun does not provide, sell, license, or lease any of the Content other than those specifically identified as being provided by Sun.

Copyright 1994-2002 Sun Microsystems, Inc

Conformité aux termes d'utilisation des documents disponibles sur le site web de Geopalan Suresh Raj pour l'utilisation de l'article de Raj, Geopalan Suresh, "A Detailed Comparaison of CORBA, DCOM and Java/RMI", mai 2002, Copyright © 1997-2001 Gopalan Suresh Raj, 8730, S. Country Drive, Apt #201, Oak Creek, WI 53154-3880 U.S.A. All rights reserved.

**NOTICE SPECIFIC TO DOCUMENTS AVAILABLE ON THIS WEBSITE**

Permission to use sources (such as source code and FAQs) from this server ("Server") is granted, provided that (1) the below copyright notice appears in all copies and that both the copyright notice and this permission notice appear, (2) use of such Documents from this Server is for informational and non-commercial or personal use only and will not be copied or posted on any network computer or broadcast in any media, and (3) no modifications of any Documents are made. Use for any other purpose is expressly prohibited by law, and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible.

**COPYRIGHT NOTICE.** Copyright © 1997-2001 Gopalan Suresh Raj, 8730, S. Country Drive, Apt #201, Oak Creek, WI 53154-3880 U.S.A. All rights reserved.



## Glossaire des acronymes utilisés

<u>Acronyme</u>	<u>Description</u>
ABC	Activity Based Costing
AEDM	Air Environment Data Model
AEMIS	Army Establishment Management Information System
AIME	Army Integrated Management Environment
ARAM	Army Risk Assessment Model
ASC	Accredited Standards Committee
ATCCIS	Army Tactical Command and Control Information System
ATS	Athene Tactical System
B2B	Business to Business
CBD	Component Based Development
CDM	Conceptual Data Model
CDR	Common Data Representation
CFCS	Combined Forces Command System
CFSS	Canadian Forces Supply System
CFSS(U)	Canadian Forces Supply System (Upgrade)
ChemXML	Chemical XML
CI	Contrôle d'installation « en anglais FC »
CIDX	Chemical Industry Data eXchange
CMS	Communication Management System
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CPES	Cellule de planification et d'exploitation du système « en anglais SEP »
CSI	Common Secure Interoperability
CSO	Contrôle de système opérationnel « en anglais OCS »
DATA PUMP	Une application logiciel générique créée par le Groupe CGI qui extrait et transforme des données dans un format XML
DCE	Distributed Computing Environment
DCIT	Direction des communications et de l'information – terrestre
DCOM	Distributed Component Object Model
DIHRS	Defence Integrated Human Resource System (COTS by PeopleSoft)
DOM	Document Object Model
DSS	Decision Support System
EBB	Electronic Battle Box
ebXML	Electronic Business XML
EDI	Electronic Data Interchange
FC	Facility Control « en français CI »
FMAS	Financial Management and Accounting System (built using SAP software)
FN (Forme normale)	La théorie des formes normales propose des critères qui permettent de juger si la redondance a été complètement éliminée d'une base de données.
FT	Forces terrestres
FTP	File Transfer Protocol
GAO	General Accounting Office (US)
GUI	Graphical User Interface
HR-XML	Human Resources XML
IDL	Interface Definition Language
IER	Information Exchange Requirements
IIOP	Internet Inter-ORB Protocol
IM	Information Management
ITL	Integration and Transformation Layer
ITS	Integrated Transaction Service
JAVA	Langage de programmation distribué par Sun Microsystem
JRMP	Java Remote Method Protocol

## Glossaire des acronymes utilisés

LegalXML	Vocabulaire XML pour l'échange de données légales
LF	Land Forces
LFCS	Land Force Command System
LFDM	Land Forces Data Model
LS-ACIS	Land Staff - Activity Coordination Information System
MathML	Mathematic XML
MIP	Multilateral Interoperability Programme
MSDM	Maintenance Support Data Model
NACHA	National Automated Clearing House Association (US)
OASIS	Organization for the Advancement of Structured Information Standards
OBS	Organization Breakdown Structure
OCS	Operational System Control
OCSP	Online Certificate Status Protocol
ODB	Operational Data Base
ODS	Operational Data Store
OMG	Object Management Group
OPERA	Operational Planning Environment and Reference Application
ORB	Object Request Broker
ORBAT	Order of Battle – Une structure d'organisation qui inclus des unités, du personnel ou de l'équipement
PDALF	Positional Data Awareness Land Forces
PDM	Physical Data Model
PKI	Public Key Infrastructure
PM	Performance Measurement
PMF	Performance Measurement Framework
POA	Portable Object Adapter
POP	Post Office Protocol
RMI	Remote Method Invocation
RosettaNet	Normes XML pour les technologies électroniques de pointe
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SAML	Security Assertion Markup Language
SAS	Situational Awareness System
SAX	Simple API for XML
SEP	System Executive and Planning
SIDES	Staffing Industry Data Exchanges Standards
SMTP	Simple Mail Transfert Protocol
SOAP	Simple Object Application Protocol
SQL	Standard Query Language
SSL	Secure Socket Layer
STCCC	Système Tactique de communication pour le Commandement et Contrôle
TMHS	Tactical Message Handling System
TranXML	Transport XML
UN/CEFACT	United Nations Centre For Trade Facilitation and Electronic Business
VPN	Virtual Private Network
XBRL	Extensible Business Reporting Language
XKMS	XML Key Management Specification
XML	Extensible Mark-up Language