

U.F.R. Sciences et Techniques Mathématiques, Informatique et Automatique

École Doctorale IAEM Lorraine

Département de Formation Doctorale Automatique

Annexe de la Thèse

Présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy I

en Automatique, Traitement du Signal, Génie Informatique

par **Jean-Philippe AUZELLE**

PROPOSITION D'UN CADRE DE MODÉLISATION MULTI-ÉCHELLES D'UN SYSTÈME D'INFORMATION EN ENTREPRISE CENTRÉ SUR LE PRODUIT.

Soutenue publiquement le 11 mars 2009 devant la commission d'examen :

Membres du jury :

Rapporteurs :	M. David Chen	Professeur, Université Bordeaux 1, Président du jury
	M. Angel Ortiz Bas	Professeur, Université Polytechnique de Valencia
Examineurs :	M. Dominique Luzeaux	Directeur adjoint de l'UMESIO à la DGA (Délégation Générale pour l'Armement), HdR
	M. Gérard Morel	Professeur, UHP Nancy I (Directeur de thèse)
	M. Hervé Panetto	Professeur, UHP Nancy I (Codirecteur de thèse)
	Mme. Frédérique Mayer	Maître de conférences, INPL (Codirectrice de thèse)
Invité :	M. François Vernadat	Professeur, Université Paul Verlaine de Metz

SOMMAIRE

Annexe.1.	Gantt du projet eProduction	4
Annexe.2.	Diagramme d'activité UML d'Enchaînement des objets de modélisation de la MOA pour notre cas d'étude	5
Annexe.3.	Meta-modèle du processus.....	6
Annexe.4.	Concepts et représentations symboliques de MEGA Modelling Suite.....	7
Annexe.5.	Diagramme d'objectifs et d'exigences du SAF global.....	11
Annexe.6.	Diagramme d'environnement de processus du SAF global.....	11
Annexe.7.	Prototype d'Outilage de modélisation	12
	Modification du méta modèle de MEGA suite.....	12
	Modification du menu contextuel de MEGA	14
	Exemple de macro pour la création d'un répertoire pour un SdS (bénéficiaire): « CopierRep&Fichier_SDS »	16
	Exemple de macro pour la création d'un répertoire pour un système contributeur :	17
	modification du portail de mega.....	19
	Code HTML du portail	19
	Creation d'un graphique de navigation avec Graphviz	22
	Création de page web pour chaque Zachman (composant-composite).....	24
	Visuel.....	24
Annexe.8.	Cahier des charge partiel « e-Production » de l'AIPL.....	25
Annexe.9.	Modèle MESA.....	29
Annexe.10.	Modèle fonctionnel inspiré du Purdue reference model	30
Annexe.11.	Création du fichier XSLT	31
	Ingénierie	31
	Résultat	32

ANNEXE.1. GANTT DU PROJET EPRODUCTION

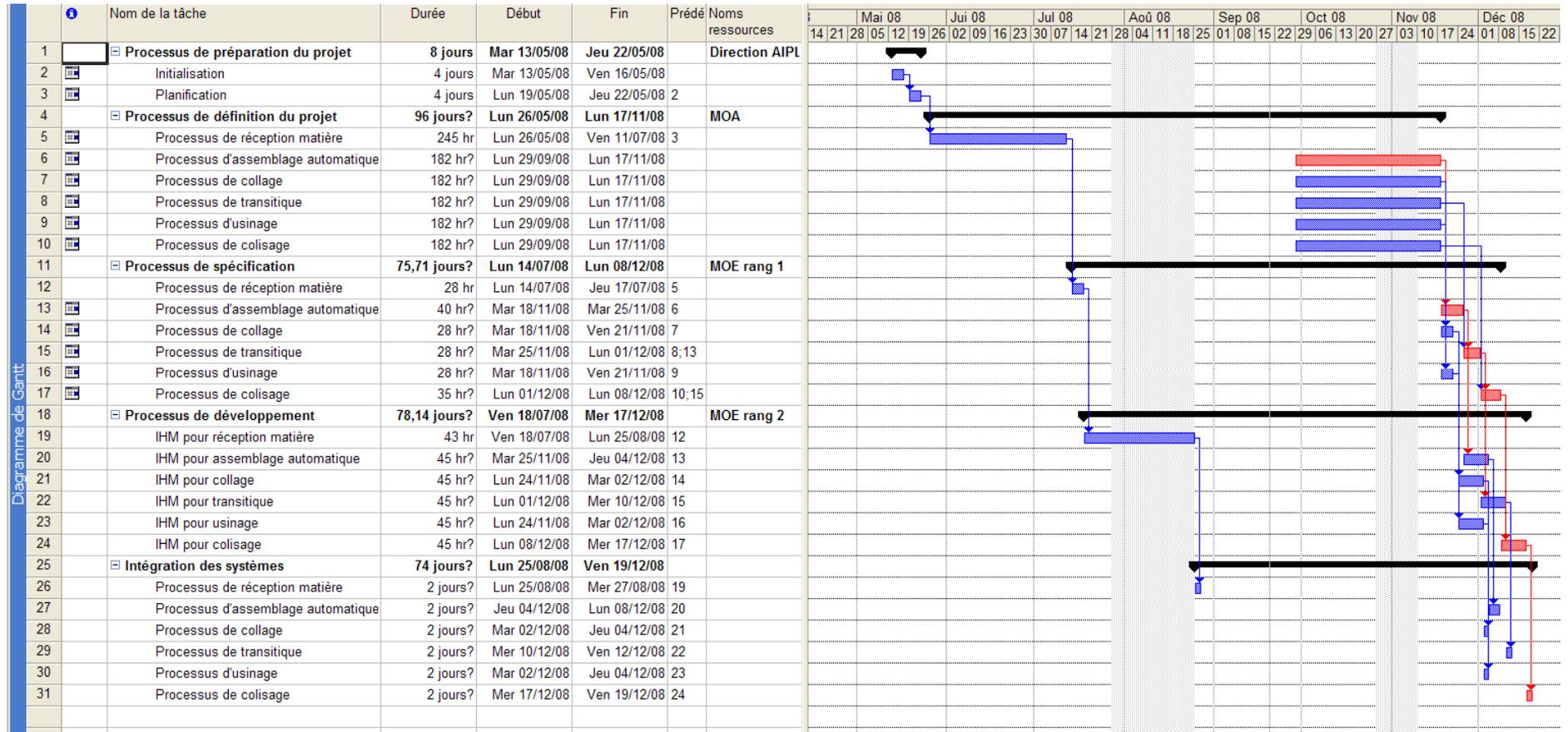
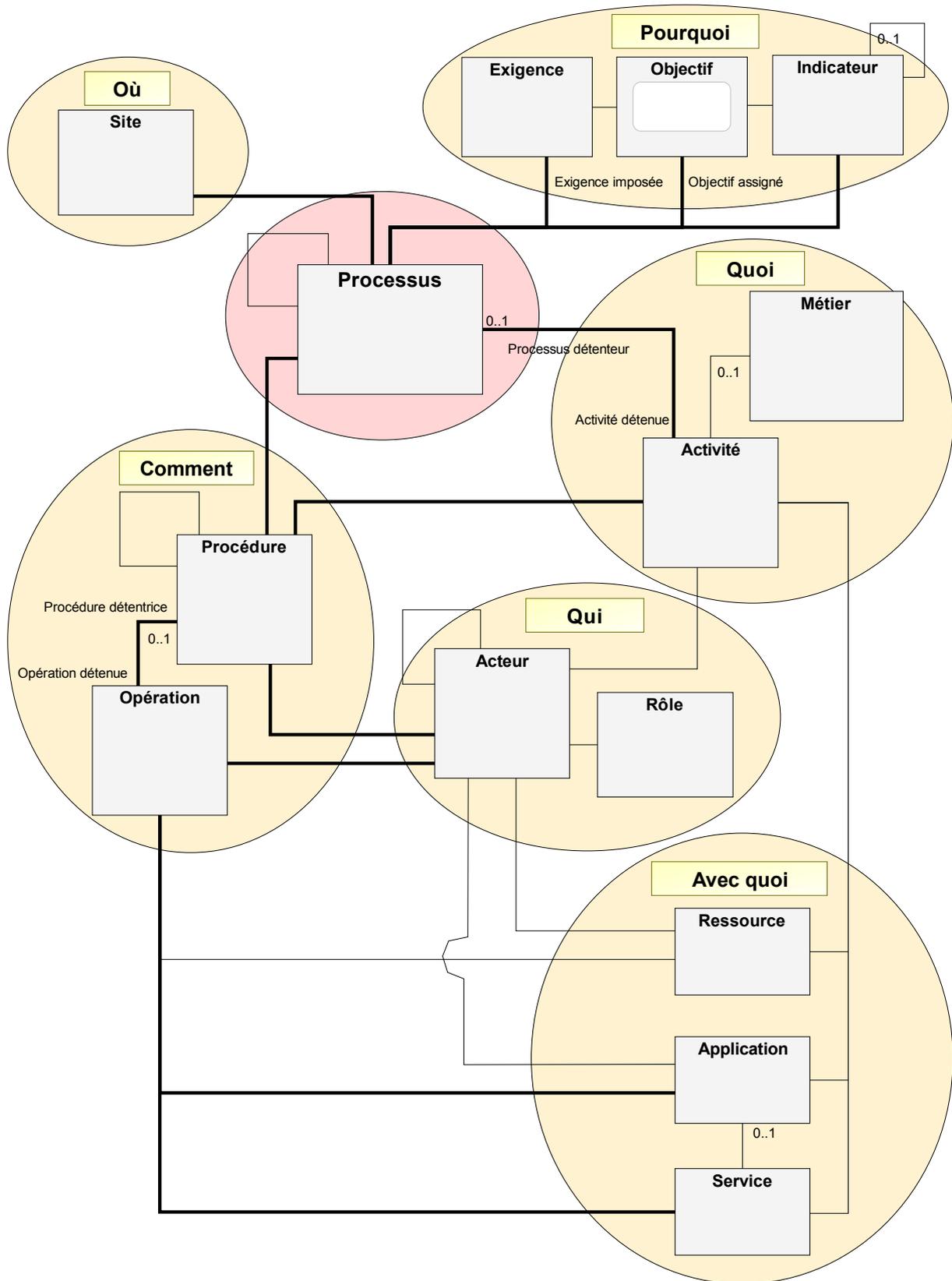


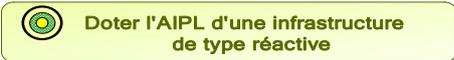
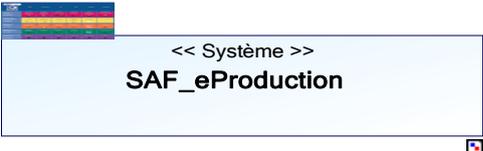
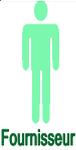
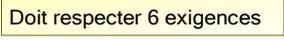
Figure 1 Gantt du projet eProduction

MEGA Process::Autour de processus



ANNEXE.4. CONCEPTS ET REPRÉSENTATIONS SYMBOLIQUES DE MEGA MODELLING SUITE

Tableau 1. Symbole - concept - définition des objets de modélisation sous MEGA Modelling Suite

Symbole	Concept	Définition
Diagramme d'objectifs et d'exigences		
	Projet	Un projet consiste en un ensemble de tâches confiées à une équipe, qui transforment un système, ou une partie d'un système, en vue d'atteindre un objectif donné.
	Objectif	Un objectif est un but que l'on cherche à atteindre ou la cible visée pour un processus ou une opération. Il permet de mettre en évidence les points que l'on veut améliorer pour ce processus ou cette opération.
	Exigence	Une exigence est un besoin ou une attente formulé explicitement, imposés comme une contrainte à respecter dans le cadre d'un projet de certification, d'organisation ou de modification du système d'information d'une entreprise.
Diagramme de paquetage		
	Paquetage	Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.
Diagramme de processus		
	Processus	Un processus est une chaîne de valeur fournissant un bien ou un service à un client interne ou externe à l'entreprise. Cette chaîne de valeur est décrite par une séquence d'activités de transformation. Elle est mise en oeuvre par des procédures.
	Acteur externe	Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.
	Message	Un message représente un flux circulant à l'intérieur de l'entreprise ou échangé entre l'entreprise et son environnement. C'est généralement un flux d'information comme une commande ou une facture. Par commodité, un flux financier comme le règlement du client, ou un flux de matière comme la livraison d'un produit est également représenté par un message.
	Note	Une note est une annotation ou remarque apportant un commentaire, un éclaircissement sur un texte ou un dessin. Les notes peuvent être de différents types : Remarque, Question, Documentation, Evolution, Correction, Anomalie, Spécification Technique, ...
Diagramme de site		
	Site	Un site est un lieu géographique où est implantée l'entreprise. Les sites peuvent être des sites-types tels que le siège, l'agence, l'usine, ou des lieux géographiques précis comme l'agence de Marseille, l'usine de Poissy, etc.

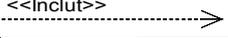
Organigramme		
 Directeur	Acteur Responsable	Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial.
 Magasinier	Acteur structure	
 Olivier NARTZ	Acteur	Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise
Diagramme de séquence		
	Instance	Un objet est une entité avec une identité et des frontières clairement définies dont l'état et le comportement sont encapsulés. Son état est défini par les valeurs de ses attributs et de ses liens avec d'autres objets. Son comportement est représenté par ses opérations et ses méthodes. Un objet est une instance de classe.
	Activation d'objet	Une activation présente la période de temps pendant laquelle un objet exécute une action, directement, ou par l'intermédiaire d'une sous-procédure. Elle représente à la fois la durée de l'action et le contrôle de l'action par l'appelant.
Diagramme de cas d'utilisation		
	Cas d'utilisation	Un cas d'utilisation est une suite d'actions qui amène un résultat observable pour un acteur particulier. Des scénarios illustrent les cas d'utilisation par l'exemple.
	Cas inclus	
	Acteur UML	

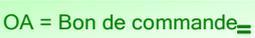
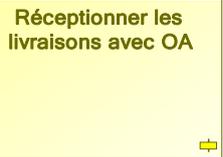
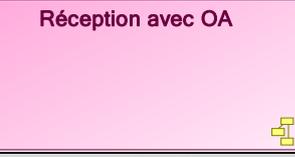
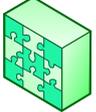
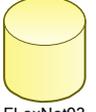
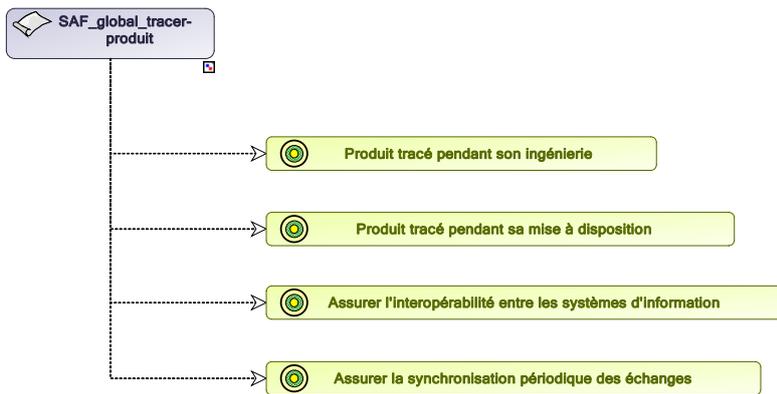
Diagramme de mise en oeuvre du processus		
 OA = Bon de commande	Note	
 Fournisseur	Acteur externe	Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.
 Réceptionner les livraisons avec OA	Opération	Une opération est une étape d'une procédure correspondant à l'intervention d'un acteur de l'organisation dans le cadre d'une des activités de l'entreprise. Ce peut être une opération industrielle comme 'usiner une pièce' ou logistique comme 'réceptionner une livraison', ou un traitement d'information comme 'enregistrer une commande'. Une opération peut être décomposée en tâches élémentaires.
 Réception avec OA	Procédure	Une procédure décrit la marche à suivre pour mettre en oeuvre tout ou partie du processus d'élaboration d'un produit ou un flux. Une procédure est représentée par une succession d'opérations déclenchées par la réception d'un message.
Diagramme d'architecture applicative		
 Flexnet	Application	Une application est un ensemble de composants logiciels qui constituent un tout cohérent au regard des développements informatiques.
 JobScheduler	Service	Un service est l'élément de découpage d'une application qui est mis à la disposition de l'utilisateur final de cette application dans le cadre de son travail. Les services représentent un découpage des applications en unités fonctionnelles élémentaires. Un service est une unité de traitement cohérente et indivisible coordonnant un ensemble de messages et d'événements afin d'effectuer une tâche dans le système d'information. Dans l'architecture applicative, le service est l'unité fonctionnelle la plus élémentaire.
 Toutes les 5 sec	Temporisateur	Un temporisateur indique le moment où se produit un événement comme le déclenchement d'une procédure ou l'émission d'un message. Un temporisateur est un événement déterminé uniquement par le temps qui s'écoule. Ex : Le lundi, à quatre heures, etc.

Diagramme d'architecture technique		
 <p>Router A IPL Borne Wi-Fi Imprimante matières</p>	Nœud	Un nœud est un élément du réseau tel qu'une imprimante, un modem, un routeur, etc. Il peut être connecté au réseau, ou bien directement à un poste de travail ou un serveur.
 <p>Ethernet A IPL</p>	Réseau	Un réseau est un système d'ordinateurs géographiquement éloignés les uns des autres, interconnectés par des télécommunications, généralement permanentes. Un réseau peut être local à un site de l'entreprise.
 <p>AKEWA</p>	Application	Une application est un ensemble de composants logiciels qui constituent un tout cohérent au regard des développements informatiques.
 <p>AIP-Lorrain</p>	Serveur	Un serveur est une ressource informatique matérielle, pouvant éventuellement disposer d'une base de données et sur laquelle des applications peuvent s'exécuter.
 <p>FLexNet93</p>	Base de données	Une base de données permet de spécifier la structure de stockage logique ou physique des données.
 <p>PDA</p>	Poste de travail (PDA)	Un poste de travail est un poste type correspondant à un profil d'utilisateur.

ANNEXE.5. DIAGRAMME D'OBJECTIFS ET D'EXIGENCES DU SAF GLOBAL



ANNEXE.6. DIAGRAMME D'ENVIRONNEMENT DE PROCESSUS DU SAF GLOBAL

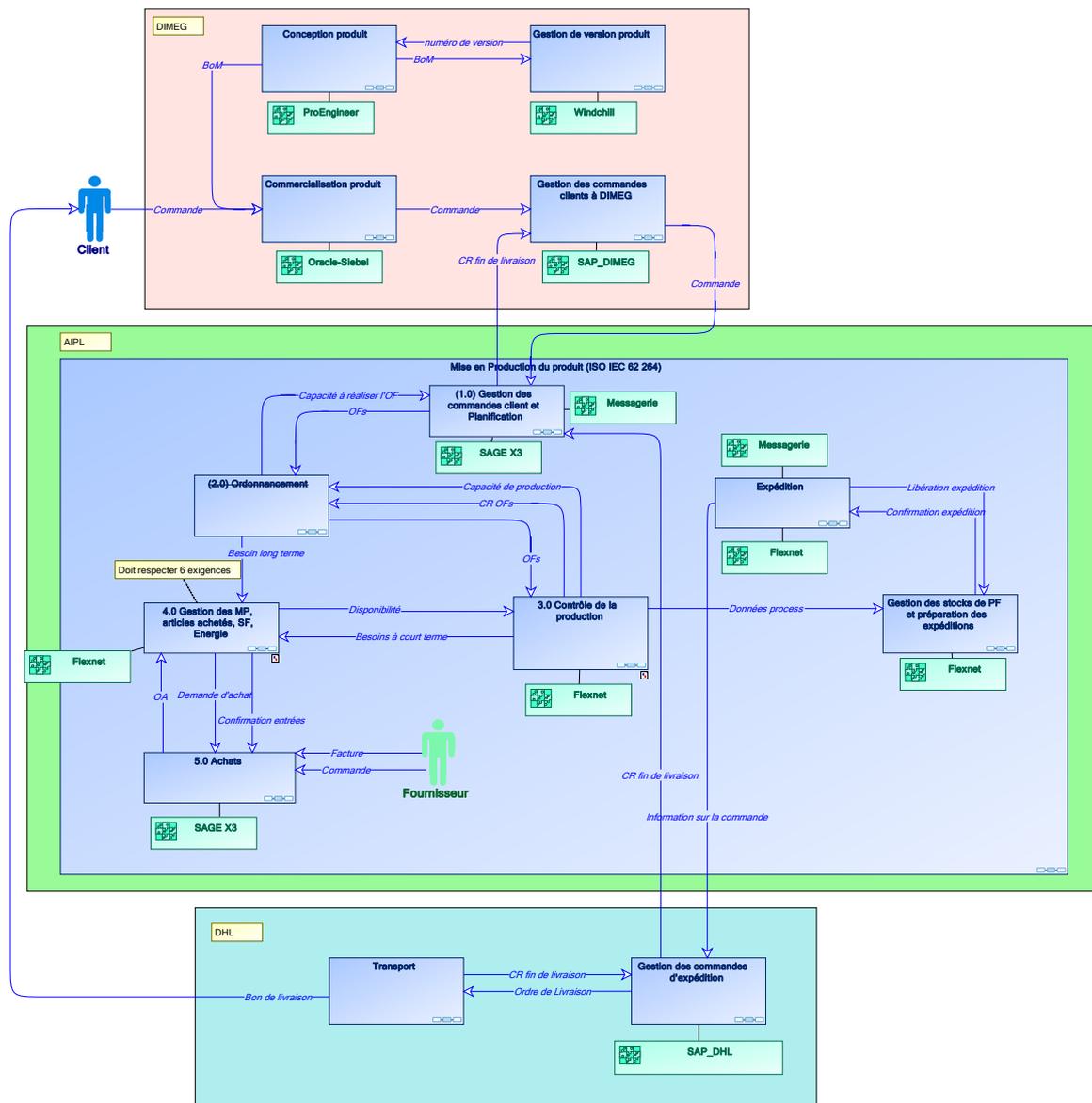


Figure 2 Diagramme d'environnement de processus du SAF global.

MODIFICATION DU MÉTA MODÈLE DE MEGA SUITE

L'expérience de notre équipe de recherche sur la meta-modélisation et l'adaptation de l'environnement MEGA Modelling Suite nous a amené, naturellement, à adapter certaines de ses fonctionnalités et surtout d'en ajouter de nouvelles. Les modifications sont relatives au méta modèle du concept de « paquetage » et du concept de « diagramme » :

- Méta classe « Paquetage »
 - Ajout du méta attribut « cellule de rattachement » de type string, longueur 63 caractères, format du méta attribut « énumération ». (Figure 3) (Figure 7)
 - Ajout du méta attribut « type de système » de type string, longueur 63 caractères, format du méta attribut « énumération ». (Figure 4)
- Méta classe « diagramme »
 - Ajout du méta attribut « cellule de rattachement » de type string, longueur 63 caractères, format du méta attribut « énumération », dont la valeur est dérivée du Paquetage détenteur de ce Diagramme. (Figure 5)

Pour simplifier le développement, s'agissant uniquement d'un prototype, nous avons implémenté le concept de PointDeVue par une liste énumérée de la combinatoire des valeurs « niveau/observation » (Figure 6).

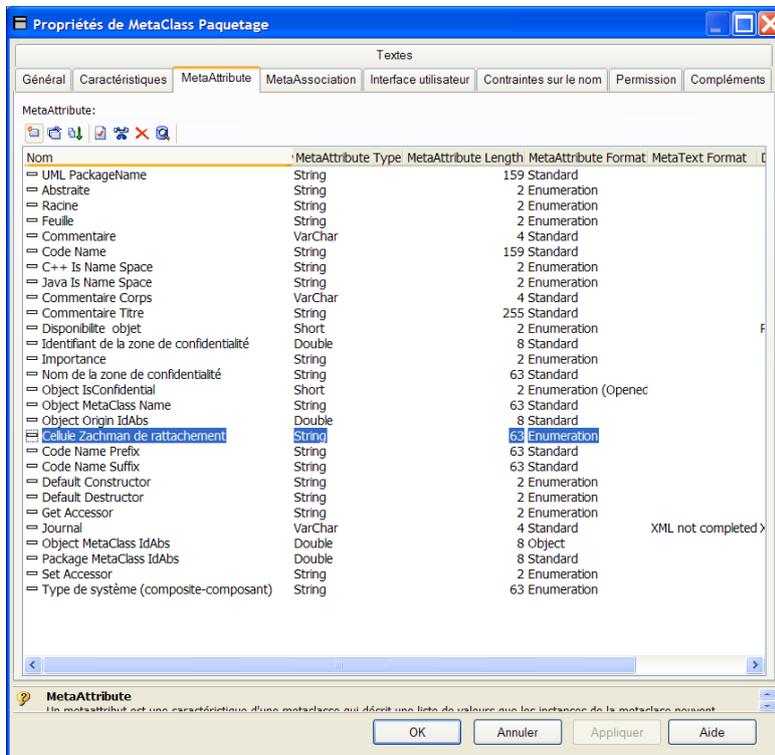


Figure 3 Propriétés de METAClass de l'objet Paquetage : méta attribut « cellule Zachman de rattachement »

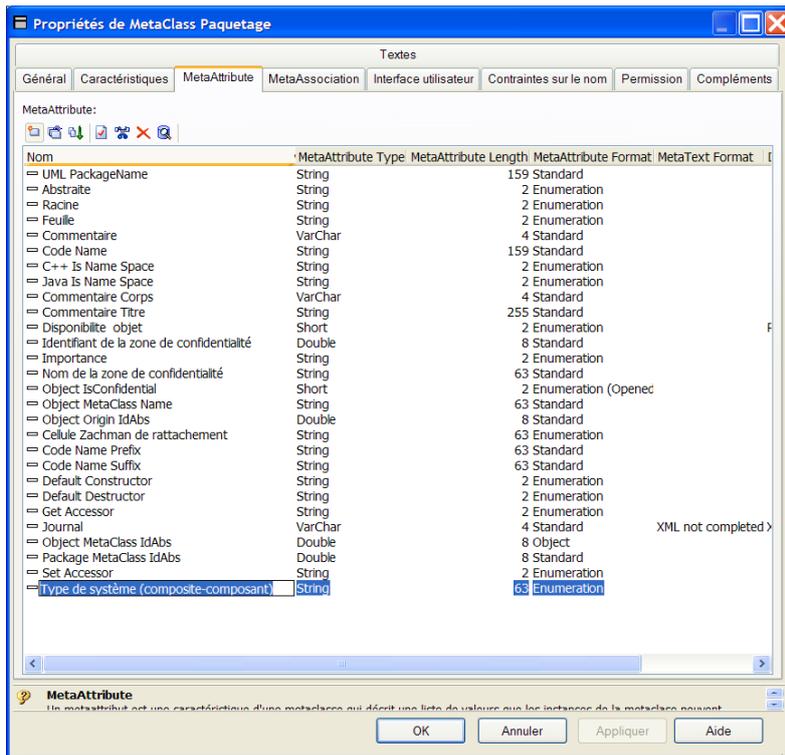


Figure 4 Propriétés de METACLASS de l'objet paquetage : méta attribut "type de système (composite-composant)"

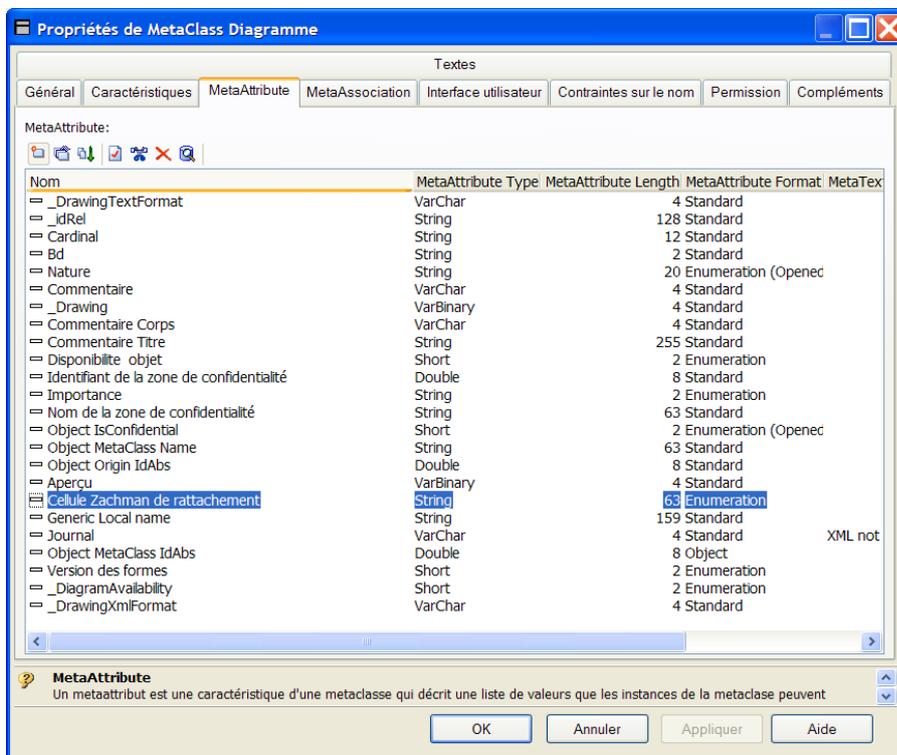


Figure 5 Propriétés de METACLASS de l'objet Diagramme : méta attribut « cellule Zachman de rattachement »

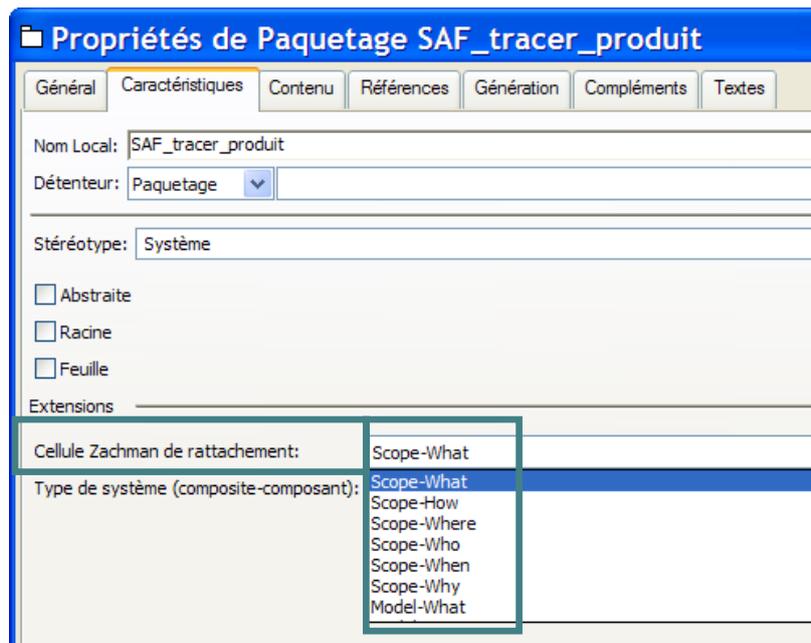


Figure 6. Stéréotype particulier pour l'association d'un paquetage ou d'un diagramme à une cellule d'un Zachman.

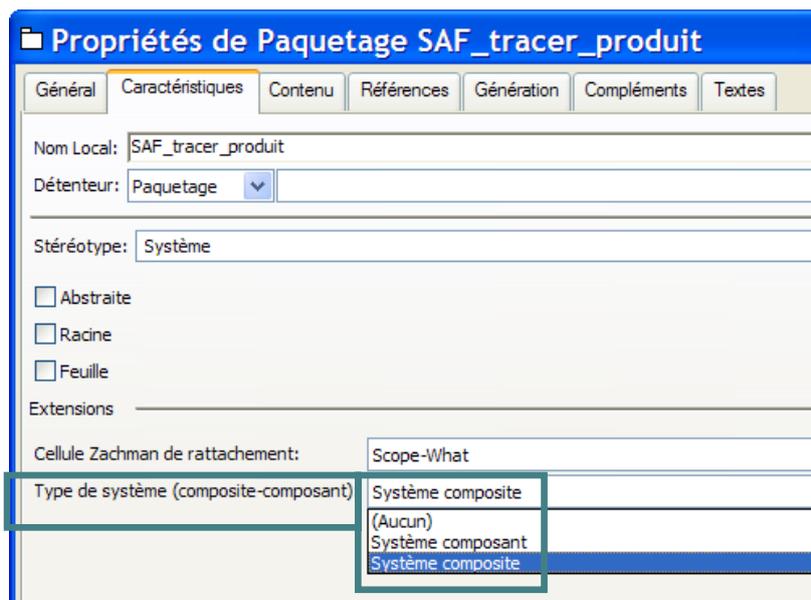
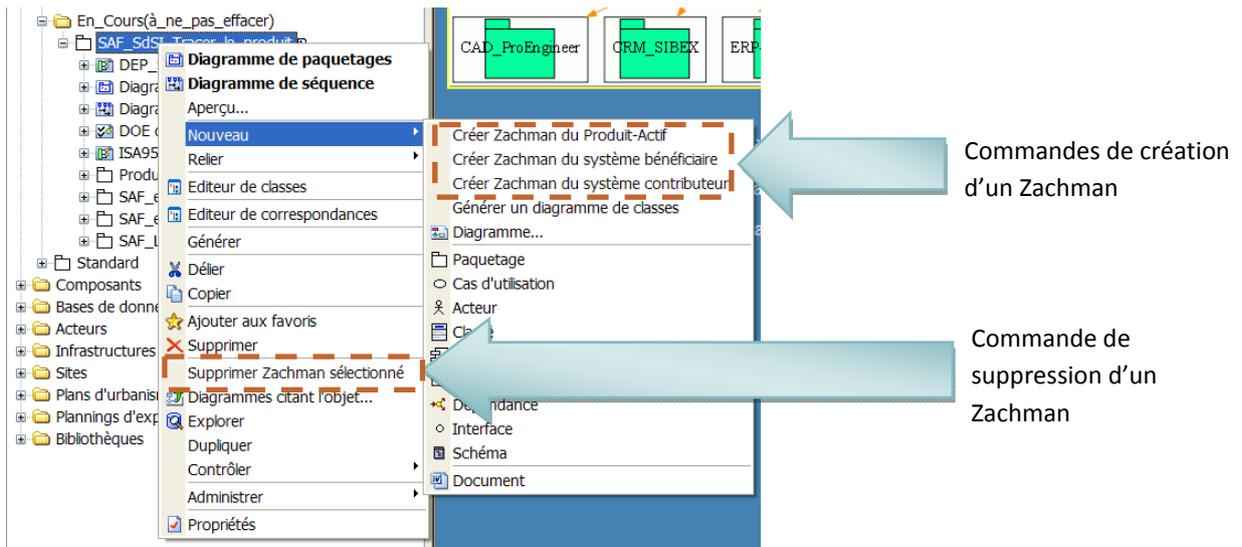


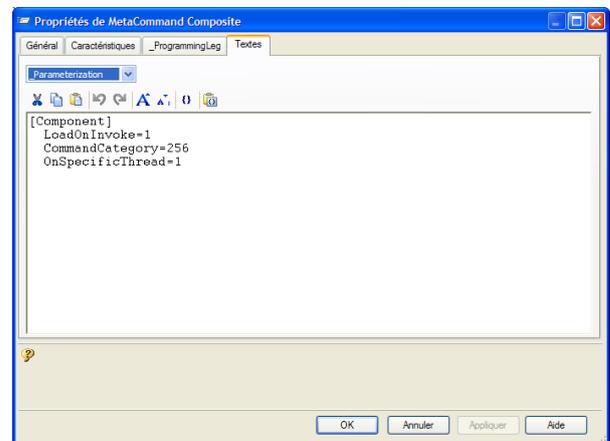
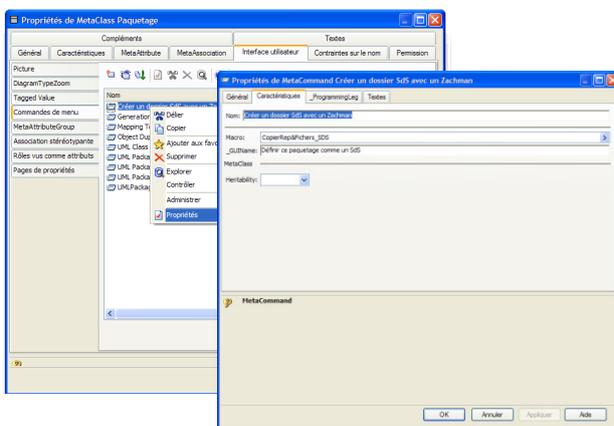
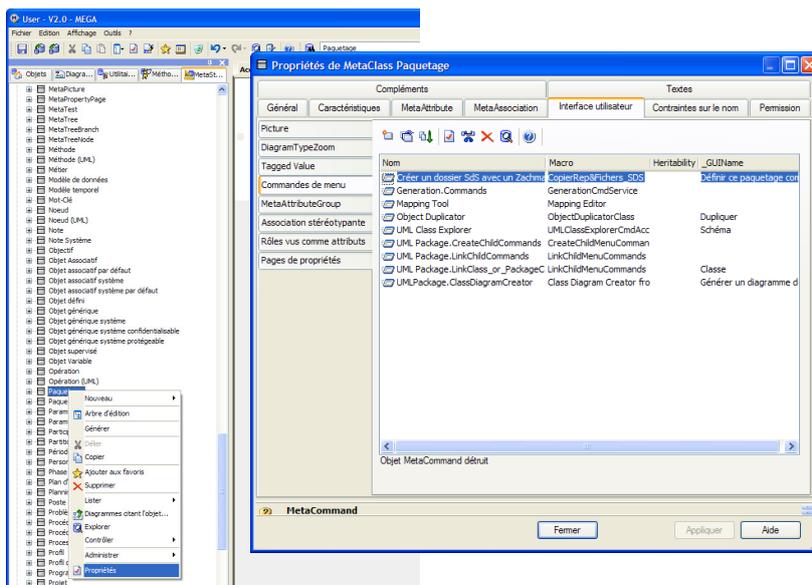
Figure 7. Stéréotype particulier pour décrire le type de système (composant / composite).

MODIFICATION DU MENU CONTEXTUEL DE MEGA

Nous avons ajouté des commandes dans le menu contextuel de l'objet « paquetage » dont l'objectif est de créer des répertoires sur le disque local du modélisateur dans lesquels seront placés les éléments de construction d'une page web incluant le Zachman correspondant à chaque système.



Pour insérer de nouvelles commandes il faut accéder (click-droit) à la méta-classe « paquetage » puis sur « propriétés ». Puis dans l'onglet « interface utilisateur » / « commande de menu » on ajoute une nouvelle commande en l'associant à une macro déjà existante. A titre d'exemple, la macro « CopierRep&Fichiers_SDS » est donnée par la suite.



Note : attention la modification du menu n'est effective que si l'on ferme le diagramme et qu'on le relance ensuite.

La valeur 256 permet d'avoir le menu dans la partie « nouveau » du menu contextuel existant.

EXEMPLE DE MACRO POUR LA CRÉATION D'UN RÉPERTOIRE POUR UN SDS (BÉNÉFICIAIRE): « CopierRep&Fichier_SDS »

Cette macro est stockée dans la base de l'environnement. Elle est accessible à partir de l'affichage « utilitaires », puis dans le repertoire « Macro VBS/MEGA/SDS ».

```
'MegaContext(External)

Option Explicit

'-----

' Copie C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Zachman

'vers

'C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Projets_User\le-nom-paquetage\

' Copie C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Zachman_vbs\sds*. *

'vers

'C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Projets_User\le-nom-paquetage\

'-----

Sub CmdInvoke(oRoot, num)

Dim fso

Dim sfol

Dim dfol

Dim oNameStereotype

Dim oStereotype

oStereotype=oRoot.GetCollection("stereotype")

If oStereotype.count > 0 Then

    oNameStereotype=oStereotype.item(1).Name

    MsgBox "votre paquetage est stéréotypé " & oNameStereotype

End If

If oNameStereotype <> "Système" Then

    MsgBox "Le stéréotype " & oNameStereotype & " doit être changé par le stéréotype 'Système'"

Else

    dfol=oRoot.GetProp("Name")

dfol=replace(dfol,":","\\")

MsgBox dfol
```

```

dfol="C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Projets_User" & dfol

sfol="C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Zachman" ' change to match the source folder path

Set fso = CreateObject("Scripting.FileSystemObject")

    If Not fso.FolderExists(dfol) Then

        fso.CopyFolder sfol, dfol

    Else

        MsgBox dfol & " already exists!", vbExclamation, "Folder Exists"

    End If

sfol = "C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Zachman_vbs\sds" ' change to match the source folder
path

Set fso = CreateObject("Scripting.FileSystemObject")

On Error Resume Next

If Not fso.FolderExists(sfol) Then

    MsgBox sfol & " is not a valid folder/path.", vbInformation, "Invalid Source"

Elseif Not fso.FolderExists(dfol) Then

    MsgBox dfol & " is not a valid folder/path.", vbInformation, "Invalid Destination"

Else

    fso.CopyFile (sfol & "\*.**"), dfol ' Change "\*.**" to "\*.xls" to move Excel Files only

End If

If Err.Number = 53 Then MsgBox "File not found"

End If

End Sub

```

EXEMPLE DE MACRO POUR LA CRÉATION D'UN RÉPERTOIRE POUR UN SYSTÈME CONTRIBUTEUR :

```

'MegaContext(External)

Option Explicit

'-----

' Copie C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Zachman

'vers

'C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Projets_User\((le-nom-paquetage)\

' Copie C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Zachman_vbs\system\*.

'vers

'C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Projets_User\((le-nom-paquetage)\

'-----

Sub CmdInvoke(oRoot, num)

Dim fso

Dim sfol

```

```

Dim sfol_vbs

Dim dfol

Dim oNameStereotype

Dim oStereotype

Dim changement

Dim chemin

chemin="C:\Program Files\MEGA\MEGA 2007\Mega_Std"

oStereotype=oRoot.GetCollection("stereotype")

If oStereotype.count > 0 Then

    oNameStereotype=oStereotype.item(1).Name

    MsgBox "votre paquetage est stéréotypé " & vbcrLf & oNameStereotype

End If

If oNameStereotype <> "Système" Then

    MsgBox "Le stéréotype " & oNameStereotype & " doit être changé par le stéréotype " & vbcrLf & "Système"

Else

    dfol=oRoot.GetProp("Name")

    dfol=replace(dfol,":","\")

    MsgBox dfol

    dfol= chemin & "\Solutions\portal\zachman\SDS\Projets_User\" & dfol

    sfol= chemin & "\Solutions\portal\zachman\SDS\Template\Zachman" ' change to match the source folder path

    sfol_vbs = chemin & "\Solutions\portal\zachman\SDS\Template\Zachman_vbs\system\" ' change to match the source folder path

    Set fso = CreateObject("Scripting.FileSystemObject")

    If Not fso.FolderExists(dfol) Then

        fso.CopyFolder sfol, dfol

        fso.CopyFile (sfol_vbs & "\*.**"), dfol

    Else

        MsgBox dfol & vbcrLf & vbcrLf & " existe déjà !", vbExclamation, "Le dossier existe"

        changement=InputBox (" Tapez 'oui'" & vbcrLf & vbcrLf & " si voulez-vous vraiment transformer " & vbcrLf & " ce Zachman en un Zachman 'composant'")

        If changement="oui" Then

            fso.DeleteFolder dfol

            fso.CopyFolder sfol, dfol

            fso.CopyFile (sfol_vbs & "\*.**"), dfol ' Change "\*.**" to "\*.xls" to move Excel Files only

        Else

            MsgBox "Il n'y aura pas de changement !"

        End If

    End If

End If

```

End If

End Sub

MODIFICATION DU PORTAIL DE MEGA

Afin d'accéder à l'outil de navigation qui donne l'accès à chaque Zachman particulier, le portail de MEGA Suite à été modifié. Un lien a été ajouté : « Ingénierie Réursive Dirigée par le Modèles ».



Figure 8. Portail de d'accueil de MEGA modifié.

CODE HTML DU PORTAIL

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Zachman r&eacute;cursif</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<script language="VBScript">
```

```
ExecuteGlobal external.megaRoot.GetObjectFromId("~2|)pZrjj1X40[MEGA Utilities]").GetProp("~07u)W)F8xO00[VB Script]")
```

```
ExecuteGlobal external.megaRoot.GetObjectFromId("~mPntwkm1XpE[MEGA Web Utilities]").GetProp("~07u)W)F8xO00[VB Script]")
```

```

</script>
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=MM_reloadPage; }}
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
//-->
</script>
</head>
<body bgcolor="#4482B4" text="#FFFFFF" link="#FFFF33" vlink="#99FF33" alink="#FF0000" language="vbscript" >
<form name="form1" method="post" action="">
<select name="graform" id="graform">
  <option value="dot" selected>dot</option>
  <option value="circo">circo</option>
  <option value="fdp">fdp</option>
  <option value="neato">neato</option>
  <option value="twopi">twopi</option>
</select>
  Rafka&icirc;chir la page si vous changez de forme du graphique (clique droit
  + refresh)
</form>
<p>&nbsp;</p>
<p>
  <script language="VBScript">
Const ForReading = 1, ForWriting = 2
Dim racine, chemin2, f
chemin2="C:/Program Files/MEGA/MEGA 2007/Mega_Std/Solutions/porta/zachman/SDS/Projets_User/"
function printNode(packageDetenu)
  Dim url, name, oStereotype, fso, typeSys, imagePackage
  name = packageDetenu.GetProp("Nom court")
  typeSys = packageDetenu.GetProp("Type de système (composite-composant)")
  oStereotype=packageDetenu.GetCollection("stereotype")

```

```

If oStereotype.count > 0 Then
    oNameStereotype=oStereotype.item(1).Name
End If

Set fso = CreateObject("Scripting.FileSystemObject")

url = chemin2 & replace(packageDetenu.GetProp("Name"), ":::","/") & "/zachman_static.html"

Set oCollec = packageDetenu.GetCollection("Paquetage détenu")

For each i in oCollec
    name_fil = i.GetProp("Nom court")

    f.write Chr(34)+name+Chr(34)+"->" + Chr(34)+name_fil+Chr(34)+";"

    f.write printNode(i)
Next

If oNameStereotype = "Système" Then
    If fso.FileExists(url) Then
        If typeSys = "Système composant" Then
            imagePackage = "package_composant.png"
        ElseIf typeSys = "Système composite" Then
            imagePackage = "package_composite.png"
        Else
            imagePackage = "package_vide.png"
        End If

        f.write Chr(34)+name+Chr(34)+
[shapefile="+Chr(34)+imagePackage+Chr(34)+",URL="+Chr(34)+url+Chr(34)+",target=_blank];"
    Else
        f.write Chr(34)+name+Chr(34)+ " [shapefile="+Chr(34)+ "package_vide.png"+Chr(34)+"]; "
    End If
End If

end function

Set racine=external.megaroot.GetObjectFromId("vNFxYgFq7L60")

Set fso = CreateObject("Scripting.FileSystemObject")

Set f = fso.OpenTextFile("C:\PROGRA~1\MEGA\MEGA20~1\Mega_Std\SOLUTI~1\portal\zachman\SDS\Template\PORTAL~1\graf.dot",
ForWriting,true)

f.write "digraph g {"

Set oCollec = racine.GetCollection("Paquetage référencé")

For each i in oCollec
    printNode(i)
Next

```

```

f.write "}"
f.Close
Dim graf
Set graf = CreateObject("WScript.Shell")
graf.Run "C:\PROGRA~1\MEGA\MEGA20~1\Mega_Std\SOLUTI~1\portal\zachman\SDS\Template\PORTAL~1\graphique.bat
"+document.form1.graform.value,,true
document.write "<img src='graf.gif' usemap='#G'/">
Dim oFso, f2
Set oFso = CreateObject("Scripting.FileSystemObject")
Set f2 = oFso.OpenTextFile("C:\PROGRA~1\MEGA\MEGA20~1\Mega_Std\SOLUTI~1\portal\zachman\SDS\Template\PORTAL~1\graf.map",
ForReading)
Document.Write f2.ReadAll
f2.Close
</script>
</p>
<table width="100%" border="0">
<tr>
<td width="41%"> Syst&egrave;me
composant associ&eacute; &agrave; un Zachman </td>
</tr>
<tr>
<td> Syst&egrave;me
composite associ&eacute; &agrave; un Zachman </td>
</tr>
<tr>
<td> Syst&egrave;me dont
on ne ma&icirc;trise pas l'ing&eacute;nieur interne (bo&icirc;te noire)</td>
</tr>
</table>
</body>
</html>

```

CREATION D'UN GRAPHIQUE DE NAVIGATION AVEC GRAPHVIZ

Le portail de MEGA a aussi été modifié pour y ajouter un lien vers un graphique de navigation du système en cours de modélisation (Figure 8 & Figure 9). La construction de ce nouveau graphique est basée sur l'application Graphviz¹. Il est ainsi généré dynamiquement au gré de l'émergence de nouveaux systèmes donc de nouveaux

¹ <http://www.graphviz.org/>

paquetages. Ce graphique de navigation permet d'avoir une vue synthétique du « système à faire » et surtout de naviguer par des liens hypertextes sur chacun des systèmes composants ou composites et donc sur leur Zachman respectifs. Les systèmes composants et composites sont distingués par des Zachman avec des couleurs de fond différentes : système composite (couleur orange), système composant (couleur verte). Si le modélisateur rencontre un système de type « boîte noire » dont il a juste la connaissance des intrants et des extrants, le graphique affiche le paquetage du système en noir (Figure 9).

La génération automatique du graphique passe une ligne de commande dans un fichier batch.

Ligne	commande
1	CD C:\Program Files\MEGA\MEGA 2007\Mega_Std\Solutions\portal\zachman\SDS\Template\Portal_sds_pa
2	set prog="C:\Program Files\Graphviz2.20\bin\%1.exe"
3	echo %prog%
4	%prog% -Gcharset=latin1 -Tcmapx -Gbgcolor=white -Gdpi=60 - Ecolor=darkorange -Ncolor=palegolenrod -Nfontcolor=black -ograf.map -Tgif - ograf.gif -shape=crow graf.dot

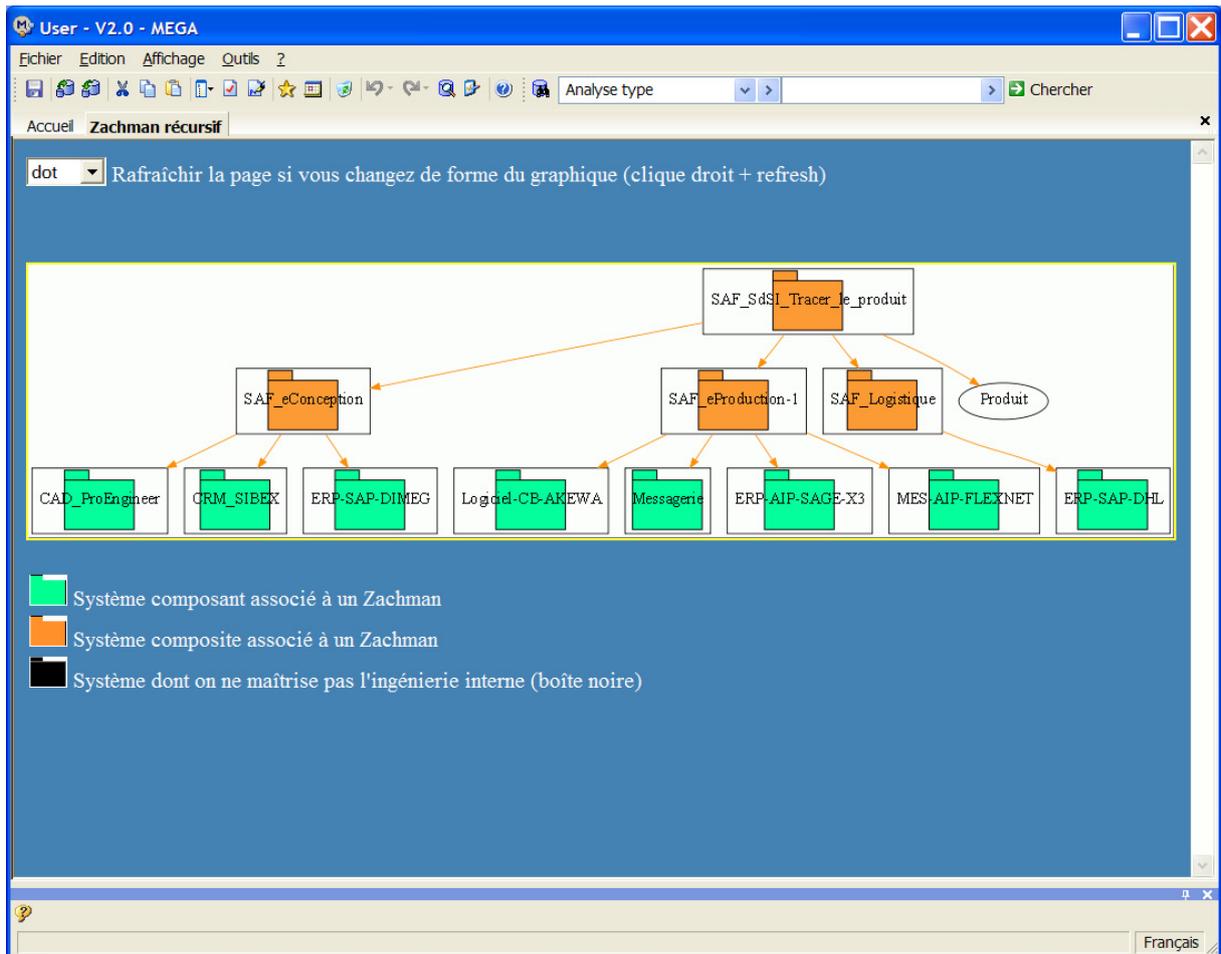


Figure 9. Graphique de navigation avec lien hypertexte vers chaque système représenté sous la forme d'un paquetage.

CRÉATION DE PAGE WEB POUR CHAQUE ZACHMAN (COMPOSANT-COMPOSITE)

VISUEL

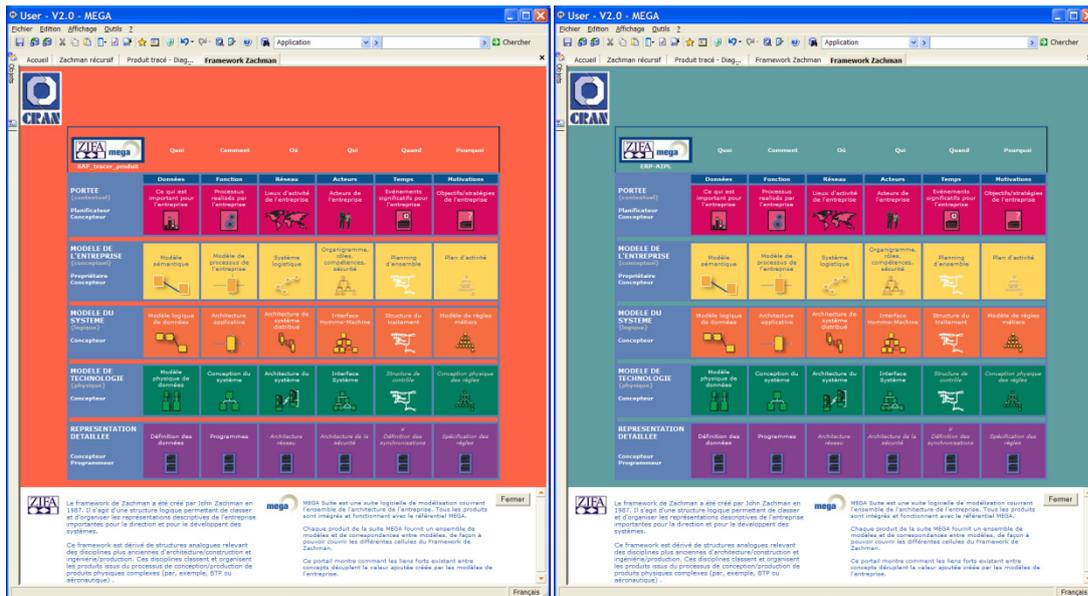


Figure 10. Cadres de Zachman particuliers associés à des système composants ou composites.

ANNEXE.8. CAHIER DES CHARGE PARTIEL « E-PRODUCTION » DE L'A IPL

Copie du contenu du document word rédigée par la direction de l'A IPL :

Projet « eFormation en eProduction »

phase " Ingénierie Produit/Process "

1. Introduction

En 2001, sous l'impulsion du Ministère de l'Éducation Nationale, a été créé le réseau de compétences AIP-PRIMECA, dans les domaines de la Conception Intégrée en Mécanique et de la Productique. Il est le résultat de la fusion entre :

- les A.I.P. (Ateliers Inter Établissements de Productique), centres de ressources régionaux utilisés comme support expérimental de formations approfondies dans le domaine de la Productique, créés en 1984 à l'initiative du Ministère de l'Éducation Nationale, en association avec le Ministère de l'Industrie et les Régions concernées ;
- le réseau PRIMECA (Pôle de Ressources Informatiques pour la MECANique), créé en 1991 à l'instigation du Comité d'Application de l'Académie des Sciences pour promouvoir l'utilisation des outils informatiques dans la conception des produits mécaniques et créer une filière de formation en Conception Mécanique Assistée par Ordinateur.

Organisées autour de centres de ressources, les missions locales dynamisent la formation dans les domaines de la Conception Intégrée en Mécanique et de la Productique. L'association des compétences pédagogiques et scientifiques crée un espace de formation autour des thèmes de l'ingénierie intégrée, favorisant la mise en place aussi bien d'enseignements approfondis par discipline que d'enseignements par projets

interdisciplinaires pour former les futurs cadres du monde industriel. Ce sont des lieux de transfert de la recherche vers la formation et l'entreprise, utilisant du matériel de dimension industrielle.

Dans chaque pôle, des moyens de type industriel permettent de former des spécialistes :

- Ayant une approche globale du système de conception intégrée et de production ;
- Maîtrisant les méthodes et techniques de modélisation, d'analyse, de production,
- d'informatisation, d'automatisation ;
- Capables de s'intégrer dans des activités pluridisciplinaires ;
- Conscients de la nécessaire cohérence des différentes fonctions de l'entreprise.

Le réseau national compte actuellement neuf pôles régionaux¹. En Lorraine, le pôle AIPPRIMECA2 est un service commun aux trois établissements suivants:

- L'Université Henri Poincaré, Nancy 1 (tutelle administrative) ;
- L'Institut National Polytechnique de Lorraine ;
- L'École Nationale Supérieure des Arts et Métiers, centre de Metz.

Le pôle AIP-PRIMECA Lorraine est depuis juin 2003 engagé dans une démarche qualité, après un audit à blanc réalisé en octobre 2004, l'audit de certification ISO 9001 version 2000 est planifié avec l'A.F.A.Q. en avril 2005.

L'AIP Lorrain, site AIP-PRIMECA Lorraine de Nancy, a toujours été conçu pour reproduire en grandeur réelle une entreprise de production manufacturière. Disposant d'outils allant de la modélisation d'entreprise, à l'usinage, l'assemblage et le contrôle tridimensionnel, en passant par la conception, l'automatisation, la robotique, la visionique, et la gestion d'entreprise, l'objectif principal du pôle est d'offrir à ses utilisateurs des matériels, des logiciels et leurs intégrations au plus proche des réalités industrielles. Le fort aspect mutualisation des moyens se traduit par une activité importante et régulière, autour de 140 000 heures élèves dispensées à près de 1000 étudiants durant l'année universitaire 2003-2004. C'est dans ce cadre que ce projet a vocation à compléter et moderniser l'offre du pôle AIP-PRIMECA Lorraine.

2. Contexte et besoins des formations en « eFormation en eProduction »

Le partage de données est une réalité au travers de l'Internet, il permet à différents intervenants de travailler en commun sur les mêmes données, c'est la notion de travail collaboratif. Déjà bien développé en phase de conception « Ingénierie collaborative » (collaborative engineering), il s'impose progressivement en production « Production collaborative » (collaborative manufacturing). En production, comme ailleurs, le travail collaboratif doit permettre de faire travailler ensemble les différents acteurs d'un projet. Par ailleurs, le recours de plus en plus fréquent à l'externalisation de la production (outsourcing) favorise son développement. En effet, confier tout ou partie de la fabrication de ses produits à un sous-traitant est aujourd'hui une pratique industrielle admise qui nécessite une organisation favorisant le partage des données de conception, de fabrication et de suivi.

La nécessité de réactivité des entreprises aux besoins des clients a rendu obligatoire une approche plus globale et en particulier à s'intéresser aux liens entre la conduite locale des procédés et la gestion locale ou distante d'ateliers et d'entreprises. Par exemple, des progiciels de gestion d'atelier basés sur le concept de MES (Manufacturing Execution System) ont émergés et ainsi permettent la mise en oeuvre du concept de travail collaboratif en production.

La plupart des industries sont sujettes à ces changements culturels et techniques (Mécaniques, Electroniques, Agroalimentaires, Pharmaceutiques, ...) et ce quelque soit le type de procédé (Continu, Batch, Manufacturier).

En particulier, un enjeu industriel important est d'assurer la traçabilité électronique du cycle de vie produit dans un contexte réglementaire toujours plus prépondérant.

Les filières utilisatrices du pôle AIP-PRIMECA Lorraine (Cycles ingénieurs ENSAIA, ENSAM, ENSEM, ENSGSI, ENSTIB, ESIAL, ESSTIN - Nouvelles filières (LMD) Licences, Licences professionnelles et Masters du domaine - IUT GMP, ...) forment des techniciens et des cadres de l'industrie dans différents secteurs industriels. Elles sont donc directement concernées par ces évolutions. L'AIP Lorrain doit, pour ce faire, proposer des manipulations mettant en évidence ces concepts déclinés suivant les aspirations des différentes filières : des techniciens en production manufacturière, aux cadres en gestion d'entreprise, en passant par les cadres de production.

L'évolution de l'AIP Lorrain dans ce contexte a débutée par l'acquisition d'un tour à commande numérique en 2000 (Rosilio TBI 540). Elle c'est poursuivie en 2003, par la rénovation de la cellule flexible d'assemblage (SFP Siemens) et l'installation d'un progiciel de gestion intégrée d'entreprise de type ERP (Adonix). Avec l'aide de la Région Lorraine sur 2003 et 2004, une plate-forme de télémaintenance (TELMA) a été développée dans cette même logique, en collaboration avec une jeune entreprise Lorraine (Predict). Parallèlement, la chaîne de conception de produits et de systèmes a été renforcée autour de la suite logicielle CATIA. En 2003, avec l'aide de la Région Lorraine, un projet, de différentes filières de l'UHP, visant à uniformiser les pratiques et les outils utilisés pédagogiquement en métrologie a permis de rénover les matériels et logiciels et d'en permettre l'intégration dans l'environnement numérique de CFAO CATIA. En 2004, le pôle a mis en avant les thèmes « Numérisation et eTraçabilité », soutenue par la région Lorraine ce projet est en cours de déploiement. Simultanément l'étude du renouvellement du centre d'usinage, âgé de 20 ans, a débouché sur un appel d'offre pour se doter d'une machine moderne de type Usinage à Grande Vitesse. Disponible depuis peu, cet investissement très important permet de boucler la chaîne numérique disponible : CAO, FAO, MOCN, MMT et Numérisation. Tous ces équipements ou logiciels sont pourvus de moyens de communication évolués et évolutifs, basés sur des Technologies de l'Information et de la Communication comme des SGDT, Ethernet et Internet.

Les nouveaux besoins pédagogiques, issus des réflexions menées dans le cadre de la mise en place du LMD, confortent la dynamique actuelle du pôle. La problématique est clairement centrée autour des concepts d'intégration d'applications d'entreprise (ERP, MES, GMAO, CAO, ...) depuis les niveaux "Business" jusqu'au niveau "Production" (B2M : Business To Manufacturing"). En particulier, les besoins en ingénierie système5 doivent être confortés, le poids de cette discipline dans la formation des futurs cadres étant prépondérantes. Dans ce sens, une application fournissant des espaces collaboratifs est en cours d'évaluation avant son déploiement à la rentrée 2005 pour la gestion des projets d'étudiants.

3. Objectifs du projet « eFormation en eProduction»

Il s'agit de doter l'AIP Lorrain d'une infrastructure de type entreprise réactive (agile), ainsi elle doit pouvoir par exemple :

- Fournir aux gestionnaires des indicateurs fiables (temps passés, prix de revient, ...);
- Fournir aux commerciaux des informations très précises nécessaires à l'acceptation des commandes (disponibilités, délais, coûts, ...);
- Mettre en fabrication rapidement un nouveau produit (conception, procédé, formation, ...)
- Réordonnancer facilement et rapidement la production (commande imprévue, panne, ...)
- Diagnostiquer rapidement des dérives (temps passés, qualité, maintenance préventive, ...);
- Répondre aux exigences des contraintes réglementaires (qualité, traçabilité,

- généalogie, ...);
- ...

Dans ce contexte, l'AIP Lorrain doit avoir le comportement d'une entreprise manufacturière, intégrant les processus de conception et de réalisation de produit usinés et/ou assemblés. Les produits sont réalisés à la demande à l'unité ou en petites séries. La conception est classique ou de type rétro-conception. Le procédé de production doit exploiter les moyens de fabrication de l'AIP Lorrain : débit matière (stock, scie), tournage (tour Rosilio TBI), fraisage (centre d'usinage DMG), contrôle (MMT DEA) et assemblage (SFP Siemens) mais aussi ceux d'autres établissements ou sites géographiques (ENSAM Metz, ...).

Différents scénarios de gestion seront disponibles, gestion locale ou distante en particulier à partir des écoles délocalisées et plus ou moins éloignées (ENSAM Metz, ENSTIB Epinal, ESSTIN Vandoeuvre, ENSGSI Nancy, ...).

La demande 2005 développe principalement le thème « Ingénierie Produit/Process », et vise à conforter deux axes principaux :

- La conception conjointe produit/process : ces solutions de fabrication numériques permettent de concevoir et de visualiser l'ensemble du processus de production d'un produit numériquement spécifié, avant la mise en service de tout matériel ou machine physique.
- L'ordonnancement réactif de la production : les outils d'ordonnancement et de modélisation en environnement complexe permettent de prendre en compte la plupart des procédés de fabrication et d'aider à la prise de décisions en comparant différentes simulations.

La figure en annexe propose une vue synthétique du projet « eFormation en eProduction ».

4. Les besoins en équipements sur l'année 2005

Les besoins en équipements pour l'année 2005, portent donc sur les points suivants :

1. Complément de la chaîne de conception numérique Produit/Process :

- Acquisition des outils logiciels de modélisation et simulation de système de production ;
- Acquisition des systèmes informatiques supports (serveur) ;
- Intégration dans l'environnement de conception produit (CAO CATIA).

2. Complément et rénovation des outils d'ordonnancement de la production :

- Mise à niveau des outils logiciels d'ordonnancement ;
- Acquisition des systèmes informatiques supports (serveur, périphériques) ;
- Intégration dans l'environnement de gestion de la production (ERP, MES, ...).

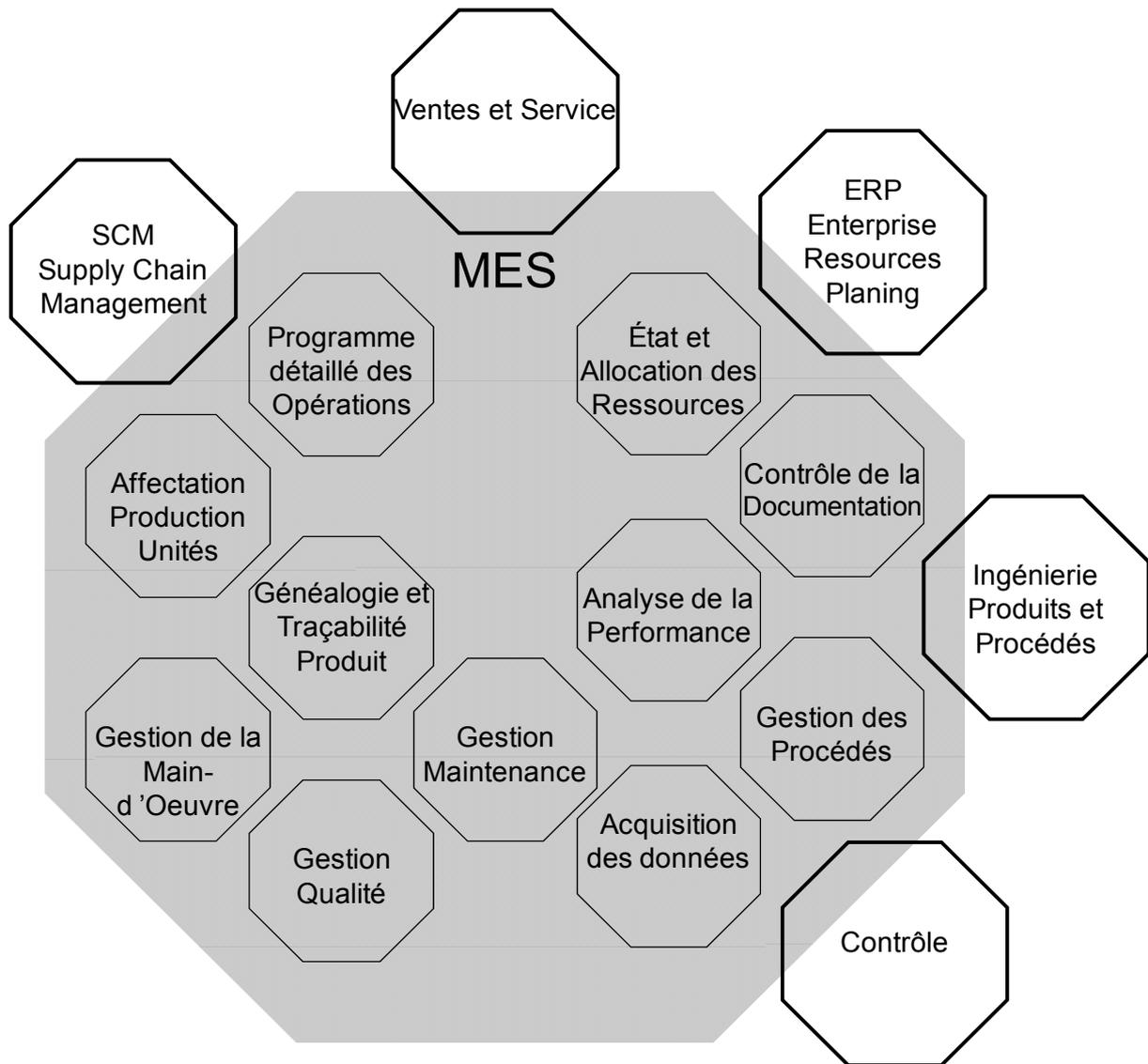


Figure 11. Modèle MESA International

ANNEXE.10. MODÈLE FONCTIONNEL INSPIRÉ DU PURDUE REFERENCE MODEL

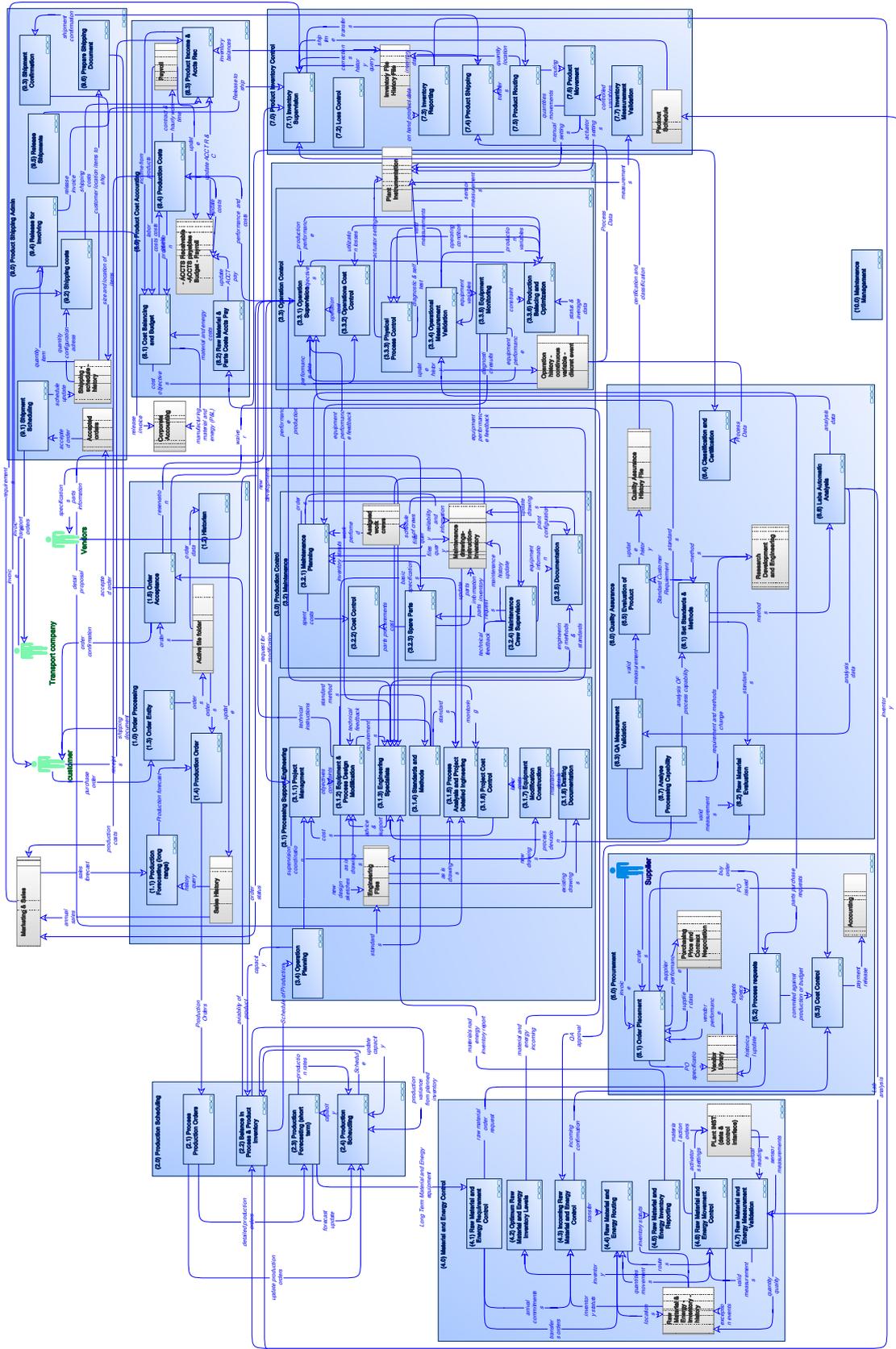


Figure 12. Modèle fonctionnel dérivé du PRM

Cette norme a été créée dans le but de faciliter l'interopérabilité sémantique entre les Systèmes de pilotage logistique (Programme de production usine ; et gestion opérationnelle) et les Systèmes de Contrôle de production (Supervision de zone, Programme détaillé de production). Les objectifs de cette norme sont :

- Établir un Vocabulaire Commun pour les procédés Continus, Discontinus & Discrets
- Trouver une Vue Commune des Problèmes
- Construire un Modèle Commun
- Définir des Structures de données
- Achever rapidement les spécifications

ANNEXE.11. CRÉATION DU FICHIER XSLT

INGÉNIERIE

Créer un dossier d'accueil quelque part sur le poste de travail.

Écrire ou récupérer un document XML au format :

```
<ADXPRODUCTS>

<ADXPRODUCT ITMREF="PRODUITTEST1" DES1AXX="Barre aluminium" STU="UN" STOFKY="C1P1"
STDFLG="1" SERMGTCOD="1" LOTMGTCOD="0"/>

<ADXPRODUCT ITMREF="PRODUITTEST2" DES1AXX="Barre plastoc" STU="UN" STOFKY="C1P1"
STDFLG="1" SERMGTCOD="1" LOTMGTCOD="0"/>

</ADXPRODUCTS>
```

Le nommer en External_ADXPRODUCTS_Output.xml (Attention, le nom de la balise root est très important !)

Ouvrir ce fichier sous XMLSpy et générer le ADXPRODUCTS.xsd correspondant.

Attention, le nom doit correspondre avec le nom de la balise root.

Récupérer les fichiers :

```
\\Aip-
lorrain\disque_c\Inetpub\wwwroot\SchemaRepository\Designs\SchemaBuilder\FlexNet\FlexNet_Products
.xml et .xsd. qui ont été générés avec XML Schema Builder d'APRISO.
```

Mapping

Ouvrir MAPFORCE

Insérer External_ADXPRODUCTS_Output.xsd

Insérer FlexNet_Products.xsd

Faire le mapping

Enregistrer le .mfd

Puis Générer le.xslt (File/Generate coding/xslt1.0

Ouvrir le .xslt généré et enlever la balise

```
<xsl:attribute name="xsi:noNamespaceSchemaLocation">  
<xsl:value-of select="'C:/DOCUME~1/Olivier.AD-UHP/Bureau/Exo/Products/FlexNet_Products.xsd'"/>  
</xsl:attribute>
```

Enregistrer sous : map_ADXPRODUCTs_FlexNetProducts.xslt

Copie des fichiers générés

Copier map_ADXPRODUCTs_FlexNetProducts.mfd sous :

```
\\Aip-lorrain\disque_c\Inetpub\wwwroot\SchemaRepository\Designs\Mapforce\FlatFile
```

Copier map_ADXPRODUCTs_FlexNetProducts.xslt sous :

```
\\Aip-lorrain\disque_c\Inetpub\wwwroot\SchemaRepository\Designs\Mapforce\FlatFile
```

Copier ADXPRODUCTS.xsd sous :

```
\\Aip-lorrain\disque_c\Inetpub\wwwroot\SchemaRepository\XMLSchemas\FlatFile
```

Préparer Flexnet pour « avaler » le fichier xml fourni

Démarrer Flexnet Configuration sur le serveur

Aller dans Mapping Configuration

Créer une nouvelle ligne :

```
FromSchema=ADXPRODUCTS  
XSLTFileName=FlatFile/map_ADXPRODUCTs_FlexNetProducts.xslt  
ToSchema=FlexNet/FlexNet_Products.xsd
```

Enregistrer

Arrêter et redémarrer tous les services FlexNet

RÉSULTAT

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
This file was generated by Altova MapForce 2008r2sp2
```

```
YOU SHOULD NOT MODIFY THIS FILE, BECAUSE IT WILL BE  
OVERWRITTEN WHEN YOU RE-RUN CODE GENERATION.
```

```
Refer to the Altova MapForce Documentation for further details.
```

```
http://www.altova.com/mapforce
```

```
-->
```

```

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" exclude-result-prefixes="xs xsi xsl">
  <xsl:include href="C:\#Fichiers\Recherche\Travail AIPL abec ON\Apriso_Functions.xsl"/>
  <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/ADXCOMMANDES">
    <FlexNet_Orders>
      <xsl:attribute name="xsi:noNamespaceSchemaLocation">
        <xsl:value-of select="'C:/DOCUME~1/Jean-Phi/Bureau/temp/FlexNet_Orders.xsd'"/>
      </xsl:attribute>
      <xsl:variable name="Vvar1_firstSource" select="."/>
      <xsl:attribute name="Version">
        <xsl:value-of select="'3.0'"/>
      </xsl:attribute>
      <xsl:variable name="Vvar19_const" select="'16'"/>
      <xsl:variable name="Vvar21_const" select="'1'"/>
      <xsl:variable name="Vvar59_const" select="'3'"/>
      <OrderHeader>
        <xsl:for-each select="GRP">
          <xsl:for-each select="FLD">
            <xsl:variable name="ExistsResult">
              <xsl:for-each select="@NAME">
                <xsl:variable name="Vvar469_NAME_string"
select="string(.)"/>
                <xsl:if test="($Vvar469_NAME_string) =
('POHNUM')">
                  <xsl:value-of select="true()"/>
                </xsl:if>
              </xsl:for-each>
            </xsl:variable>
            <xsl:variable name="Vvar467_exists" select="string-length( $ExistsResult
)&gt;0"/>
            <xsl:if test="string($Vvar467_exists)='true' or
string($Vvar467_exists)='1'">
              <xsl:variable name="Vvar473_FLD_string" select="string(.)"/>
              <OrderNo>
                <xsl:value-of select="$Vvar473_FLD_string"/>
              </OrderNo>
            </xsl:if>
          </xsl:for-each>
        </xsl:for-each>
      <OrderType>
        <xsl:value-of select="number($Vvar19_const)"/>
      </OrderType>
      <OrderStatus>
        <xsl:value-of select="number($Vvar21_const)"/>
      </OrderStatus>
      <xsl:for-each select="GRP">
        <xsl:for-each select="FLD">
          <xsl:variable name="Vvar478_FLD_string" select="string(.)"/>
          <xsl:variable name="ExistsResult1">
            <xsl:for-each select="@NAME">
              <xsl:variable name="Vvar482_NAME_string"
select="string(.)"/>
              <xsl:if test="($Vvar482_NAME_string) =
('ORDDAT')">
                <xsl:value-of select="true()"/>
              </xsl:if>
            </xsl:for-each>
          </xsl:variable>
          <xsl:variable name="Vvar480_exists" select="string-length(
$ExistsResult1 )&gt;0"/>
          <xsl:if test="string($Vvar480_exists)='true' or
string($Vvar480_exists)='1'">
            <xsl:variable name="Vvar487_result_apriso_time">
              <xsl:call-template name="apriso_time">
                <xsl:with-param name="YYYYMMDD"
select="$Vvar478_FLD_string"/>
                <xsl:with-param name="HHMMSS"
select="'000000'"/>
              </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="Vvar488_result_apriso_time_dateTime"
select="string($Vvar487_result_apriso_time)"/>
            <OrderDate>
              <xsl:value-of
select="$Vvar488_result_apriso_time_dateTime"/>

```

```

        </OrderDate>
    </xsl:if>
</xsl:for-each>
</xsl:for-each>
<ProgressStatus>
    <xsl:value-of select="number($Vvar21_const)"/>
</ProgressStatus>
<Active>
    <xsl:value-of select="string($Vvar21_const)='true' or string($Vvar21_const)='1'"/>
</Active>
<Partner>
    <xsl:for-each select="GRP">
        <xsl:for-each select="FLD">
            <xsl:variable name="ExistsResult2">
                <xsl:for-each select="@NAME">
                    <xsl:variable
name="Vvar497_NAME_string" select="string(.)"/>
                    <xsl:if test="($Vvar497_NAME_string
= ('BPSNUM'))">
                        <xsl:value-of select="true()"/>
                    </xsl:if>
                </xsl:for-each>
            </xsl:variable>
            <xsl:variable name="Vvar495_exists" select="string-length(
$ExistsResult2 )&gt;0"/>
            <xsl:if test="string($Vvar495_exists)='1'">
                <xsl:variable name="Vvar501_FLD_string"
select="string(.)"/>
                <Medium>
                    <xsl:value-of
select="$Vvar501_FLD_string"/>
                </Medium>
            </xsl:if>
        </xsl:for-each>
    </xsl:for-each>
</PartnerRelationship>
<PartnerRole>
    <xsl:value-of select="number($Vvar59_const)"/>
</PartnerRole>
<xsl:for-each select="GRP">
    <xsl:for-each select="FLD">
        <xsl:variable name="ExistsResult3">
            <xsl:for-each select="@NAME">
                <xsl:variable
name="Vvar510_NAME_string" select="string(.)"/>
                <xsl:if
test="($Vvar510_NAME_string) = ('BPSNUM')">
                    <xsl:value-of
select="true()"/>
                </xsl:if>
            </xsl:for-each>
        </xsl:variable>
        <xsl:variable name="Vvar508_exists"
select="string-length( $ExistsResult3 )&gt;0"/>
        <xsl:if test="string($Vvar508_exists)='true' or
string($Vvar508_exists)='1'">
            <xsl:variable
name="Vvar514_FLD_string" select="string(.)"/>
            <Account>
                <xsl:value-of
select="$Vvar514_FLD_string"/>
            </Account>
        </xsl:if>
    </xsl:for-each>
</xsl:for-each>
<PartnerID>
    <xsl:attribute name="NodeType">
        <xsl:value-of select=""Field"/>
    </xsl:attribute>
</PartnerID>
</PartnerRelationship>
</Partner>
<xsl:for-each select="TAB">
    <xsl:variable name="ExistsResult4">
        <xsl:for-each select="@ID">

```

```

        <xsl:variable name="Vvar522_ID_string" select="string(.)"/>
        <xsl:if test="($Vvar522_ID_string) = ('POH2_1')">
            <xsl:value-of select="true()"/>
        </xsl:if>
    </xsl:for-each>
</xsl:variable>
<xsl:variable name="Vvar520_exists" select="string-length( $ExistsResult4
)>>0"/>
    <xsl:for-each select="LIN">
        <xsl:variable name="Vvar518_LIN" select="."/>
        <xsl:if test="string($Vvar520_exists)='true' or
string($Vvar520_exists)=1">
            <OrderDetail>
                <xsl:for-each select="$Vvar1_firstSource/GRP">
                    <xsl:for-each select="FLD">
                        <xsl:variable
name="ExistsResult5">
                            <xsl:for-each
select="@NAME">
                                <xsl:variable name="Vvar533_NAME_string" select="string(.)"/>
                                <xsl:if
test="($Vvar533_NAME_string) = ('POHNUM')">
                                    <xsl:value-of select="true()"/>
                                </xsl:if>
                            </xsl:for-each>
                        </xsl:variable>
                        <xsl:variable
name="Vvar531_exists" select="string-length( $ExistsResult5 )>>0"/>
                        <xsl:if
test="string($Vvar531_exists)='true' or string($Vvar531_exists)=1">
                            <xsl:variable
name="Vvar537_FLD_string" select="string(.)"/>
                            <OrderNo>
                                <xsl:value-of select="$Vvar537_FLD_string"/>
                            </OrderNo>
                        </xsl:if>
                    </xsl:for-each>
                </xsl:for-each>
                <OrderType>
                    <xsl:value-of select="number('16')"/>
                </OrderType>
                <xsl:for-each select="@NUM">
                    <xsl:variable
name="Vvar541_NUM_byte" select="number(.)"/>
                    <OrderLineNo>
                        <xsl:value-of
select="number($Vvar541_NUM_byte)"/>
                    </OrderLineNo>
                </xsl:for-each>
                <OrderStatus>
                    <xsl:value-of select="number('1')"/>
                </OrderStatus>
                <xsl:for-each select="FLD">
                    <xsl:variable name="ExistsResult6">
                        <xsl:for-each
select="@NAME">
                            <xsl:variable
name="Vvar551_NAME_string" select="string(.)"/>
                            <xsl:if
test="($Vvar551_NAME_string) = ('UOM')">
                                <xsl:value-of select="true()"/>
                            </xsl:if>
                        </xsl:for-each>
                    </xsl:variable>
                    <xsl:variable name="Vvar549_exists"
select="string-length( $ExistsResult6 )>>0"/>
                    <xsl:if
test="string($Vvar549_exists)='true' or string($Vvar549_exists)=1">
                        <xsl:variable
name="Vvar555_FLD_string" select="string(.)"/>
                        <OrderedUomCode>

```

<pre> select="\$Vvar555_FLD_string"/> </pre>	<pre> <xsl:value-of </OrderedUomCode> </xsl:if> </xsl:for-each> <xsl:for-each select="FLD"> <xsl:variable <xsl:variable name="ExistsResult7"> <xsl:for-each <xsl:variable <xsl:if </xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable name="Vvar559_exists" <xsl:if <QuantityOrdered> <xsl:value-of </QuantityOrdered> </xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable name="Vvar559_exists" <xsl:if <QuantityOrdered> <xsl:value-of </QuantityOrdered> </xsl:if> </xsl:for-each> </xsl:variable> <xsl:for-each select="FLD"> <xsl:variable name="ExistsResult8"> <xsl:for-each <xsl:variable <xsl:if </xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable name="Vvar569_exists" <xsl:if <xsl:variable <ToFacility> <xsl:value-of </ToFacility> </xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable name="Vvar569_exists" <xsl:if <xsl:variable <ToFacility> <xsl:value-of </ToFacility> </xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable name="Vvar579_exists" <xsl:if <xsl:variable <ToWarehouse> </pre>
<pre> name="Vvar557_FLD_string" select="string(.)"/> </pre>	
<pre> select="@NAME"> </pre>	
<pre> name="Vvar561_NAME_string" select="string(.)"/> </pre>	
<pre> test="(\$Vvar561_NAME_string) = ('QTYUOM')> <xsl:value-of select="true()"/> </pre>	
<pre> select="string-length(\$ExistsResult7)>0"/> </pre>	
<pre> test="string(\$Vvar559_exists)='true' or string(\$Vvar559_exists)='1'"> </pre>	
<pre> select="number(\$Vvar557_FLD_string)"/> </pre>	
<pre> select="@NAME"> </pre>	
<pre> name="Vvar571_NAME_string" select="string(.)"/> </pre>	
<pre> test="(\$Vvar571_NAME_string) = ('PRHFCY')> <xsl:value-of select="true()"/> </pre>	
<pre> select="string-length(\$ExistsResult8)>0"/> </pre>	
<pre> test="string(\$Vvar569_exists)='true' or string(\$Vvar569_exists)='1'"> </pre>	
<pre> name="Vvar575_FLD_string" select="string(.)"/> </pre>	
<pre> select="\$Vvar575_FLD_string"/> </pre>	
<pre> select="@NAME"> </pre>	
<pre> name="Vvar581_NAME_string" select="string(.)"/> </pre>	
<pre> test="(\$Vvar581_NAME_string) = ('FCYADD')> <xsl:value-of select="true()"/> </pre>	
<pre> select="string-length(\$ExistsResult9)>0"/> </pre>	
<pre> test="string(\$Vvar579_exists)='true' or string(\$Vvar579_exists)='1'"> </pre>	
<pre> name="Vvar585_FLD_string" select="string(.)"/> </pre>	

```

select="$Vvar585_FLD_string"/>

string('1')='1'"/>

select=""Field""/>

name="ExistsResult10">
select="@NAME">
    <xsl:variable name="Vvar597_NAME_string" select="string(.)"/>
test="($Vvar597_NAME_string) = ('ITMREF')">
    <xsl:value-of select="true()"/>

name="Vvar595_exists" select="string-length( $ExistsResult10 )&gt;0"/>
test="string($Vvar595_exists)='true' or string($Vvar595_exists)='1'"/>
name="Vvar601_FLD_string" select="string(.)"/>
name="ProductNo">
    <xsl:value-of select="$Vvar601_FLD_string"/>

select="$Vvar1_firstSource/GRP">

name="Vvar605_FLD_string" select="string(.)"/>
name="ExistsResult11">
each select="@NAME">
    <xsl:variable name="Vvar609_NAME_string" select="string(.)"/>
    <xsl:if test="($Vvar609_NAME_string) = ('POHNUM')">
        <xsl:value-of select="true()"/>
    </xsl:if>
</xsl:for-each>

name="Vvar607_exists" select="string-length( $ExistsResult11 )&gt;0"/>
test="string($Vvar607_exists)='true' or string($Vvar607_exists)='1'"/>
each select="$Vvar518_LIN/@NUM">
    <xsl:variable name="Vvar614_NUM_byte" select="number(.)"/>
    <WipOrderNo>

```

```

<xsl:value-of
    </ToWarehouse>
</xsl:if>
</xsl:for-each>
<ProgressStatus>
    <xsl:value-of select="number('1')"/>
</ProgressStatus>
<Active>
    <xsl:value-of select="string('1')='true' or
</Active>
<ProductID>
    <xsl:attribute name="NodeType">
        <xsl:value-of
            </xsl:attribute>
            <xsl:for-each select="FLD">
                <xsl:variable
                    <xsl:for-each
                        <xsl:if
                            <xsl:if>
                                </xsl:for-each>
                                </xsl:variable>
                                <xsl:variable
                                    <xsl:if
                                        <xsl:variable
                                            <xsl:attribute
                                                </xsl:attribute>
                                            </xsl:if>
                                        </xsl:for-each>
                                    </ProductID>
                                    <WipOrder>
                                        <xsl:for-each
                                            <xsl:for-each select="FLD">
                                                <xsl:variable
                                                    <xsl:variable
                                                        <xsl:if
                                                            <xsl:for-

```

```

                <xsl:value-of select="concat($Vvar605_FLD_string, format-number(number($Vvar614_NUM_byte), '0000'))"/>
            </WipOrderNo>
        </xsl:for-each>
    </xsl:if>
</xsl:for-each>
</xsl:for-each>
<WipOrderType>
    <xsl:value-of
select="number('16')"/>
    </WipOrderType>
    <xsl:for-each select="FLD">
        <xsl:variable
name="Vvar624_FLD_string" select="string(.)"/>
        <xsl:variable
name="ExistsResult12">
            <xsl:for-each
select="@NAME">
                <xsl:variable name="Vvar628_NAME_string" select="string(.)"/>
                <xsl:if
test="($Vvar628_NAME_string) = ('QTYUOM')">
                    <xsl:value-of select="true()"/>
                </xsl:if>
            </xsl:for-each>
            <xsl:variable
name="Vvar626_exists" select="string-length( $ExistsResult12 )&gt;0"/>
            <xsl:if
test="string($Vvar626_exists)='true' or string($Vvar626_exists)='1'">
                <OrderQuantity>
                    <xsl:value-of select="number($Vvar624_FLD_string)"/>
                </OrderQuantity>
            </xsl:if>
        </xsl:for-each>
    </WipOrderType>
    <Priority>
        <xsl:value-of
select="number('1')"/>
    </Priority>
    <xsl:for-each select="FLD">
        <xsl:variable
name="Vvar636_FLD_string" select="string(.)"/>
        <xsl:variable
name="ExistsResult13">
            <xsl:for-each
select="@NAME">
                <xsl:variable name="Vvar640_NAME_string" select="string(.)"/>
                <xsl:if
test="($Vvar640_NAME_string) = ('EXTRCPDAT')">
                    <xsl:value-of select="true()"/>
                </xsl:if>
            </xsl:for-each>
            <xsl:variable
name="Vvar638_exists" select="string-length( $ExistsResult13 )&gt;0"/>
            <xsl:if
test="string($Vvar638_exists)='true' or string($Vvar638_exists)='1'">
                <xsl:variable
name="Vvar645_result_apriso_time">
                    <xsl:call-template name="apriso_time">
                        <xsl:with-param name="YYYYMMDD" select="$Vvar636_FLD_string"/>
                        <xsl:with-param name="HHMMSS" select="'000000'"/>
                    </xsl:call-template>
                </xsl:variable>
            </xsl:if>
        </xsl:for-each>
    </xsl:variable>
    <xsl:variable
name="Vvar646_result_apriso_time_dateTime" select="string($Vvar645_result_apriso_time)"/>

```

<pre> <ScheduledStartDate> <xsl:value-of select="\$Vvar646_result_apriso_time_dateTime"/> </ScheduledStartDate> name="Vvar648_FLD_string" select="string(.)"/> name="ExistsResult14"> select="@NAME"> <xsl:variable name="Vvar652_NAME_string" select="string(.)"/> test="(\$Vvar652_NAME_string) = ('EXTRCPDAT')"> <xsl:value-of select="true()"/> name="Vvar650_exists" select="string-length(\$ExistsResult14)&gt;0"/> test="string(\$Vvar650_exists)='true' or string(\$Vvar650_exists)='1'"> name="Vvar657_result_apriso_time"> <xsl:call-template name="apriso_time"> <xsl:with-param name="YYYYMMDD" select="\$Vvar648_FLD_string"/> <xsl:with-param name="HHMMSS" select="'000000'"/> </xsl:call-template> name="Vvar658_result_apriso_time_dateTime" select="string(\$Vvar657_result_apriso_time)/"> <xsl:value-of select="\$Vvar658_result_apriso_time_dateTime"/> name="ExistsResult15"> select="@NAME"> <xsl:variable name="Vvar664_NAME_string" select="string(.)"/> test="(\$Vvar664_NAME_string) = ('UOM')"> <xsl:value-of select="true()"/> name="Vvar662_exists" select="string-length(\$ExistsResult15)&gt;0"/> test="string(\$Vvar662_exists)='true' or string(\$Vvar662_exists)='1'"> name="Vvar668_FLD_string" select="string(.)"/> <xsl:value-of select="\$Vvar668_FLD_string"/> select="number('1')"/> </pre>	<pre> </xsl:if> </xsl:for-each> <xsl:for-each select="FLD"> <xsl:variable <xsl:variable <xsl:for-each <xsl:if <xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable <xsl:if <xsl:variable <xsl:variable <xsl:if <xsl:variable </xsl:variable> <xsl:variable <DueDate> </DueDate> </xsl:if> </xsl:for-each> <xsl:for-each select="FLD"> <xsl:variable <xsl:for-each <xsl:if <xsl:variable <xsl:if </xsl:if> </xsl:for-each> </xsl:variable> <xsl:variable <xsl:if <xsl:variable <UomCode> </UomCode> </xsl:if> </xsl:for-each> <WorkOrderStatus> <xsl:value-of </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------


```

$ExistsResult17 )&gt;0"/>
string($Vvar699_exists)=1">
select="string(.)"/>
select="$Vvar705_FLD_string"/>
<xsl:variable name="Vvar699_exists" select="string-length(
<xsl:if test="string($Vvar699_exists)='true' or
<xsl:variable name="Vvar705_FLD_string"
<OrderNo>
<xsl:value-of
</OrderNo>
</xsl:if>
</xsl:for-each>
</xsl:for-each>
<OrderType>
<xsl:value-of select="number($Vvar19_const)"/>
</OrderType>
<PartnerID>
<xsl:for-each select="GRP">
<xsl:for-each select="FLD">
<xsl:variable name="ExistsResult18">
<xsl:for-each select="@NAME">
<xsl:variable
<xsl:if
<xsl:value-of
</xsl:if>
</xsl:for-each>
</xsl:variable>
<xsl:variable name="Vvar712_exists"
<xsl:if test="string($Vvar712_exists)='true' or
<xsl:variable
<xsl:attribute name="Account">
<xsl:value-of
</xsl:attribute>
</xsl:if>
</xsl:for-each>
</xsl:for-each>
<xsl:attribute name="PartnerRole">
<xsl:value-of select="number($Vvar59_const)"/>
</xsl:attribute>
</PartnerID>
</OrderPartner>
<JobCreationEvent>
<xsl:for-each select="TAB">
<xsl:variable name="ExistsResult19">
<xsl:for-each select="@ID">
<xsl:variable name="Vvar726_ID_string"
<xsl:if test="($Vvar726_ID_string) = ('POH2_1')">
<xsl:value-of select="true()"/>
</xsl:if>
</xsl:for-each>
</xsl:variable>
<xsl:variable name="Vvar724_exists" select="string-length(
<xsl:for-each select="LIN">
<xsl:variable name="Vvar722_LIN" select="."/>
<xsl:if test="string($Vvar724_exists)='true' or
<Job>
<xsl:for-each
<xsl:for-each select="FLD">
<xsl:variable
<xsl:variable
<xsl:for-
each select="@NAME">

```



```

select="$Vvar1_firstSource/GRP">
name="Vvar773_FLD_string" select="string(.)"/>
name="ExistsResult22">
each select="@NAME">
    <xsl:variable name="Vvar777_NAME_string" select="string(.)"/>
    <xsl:if test="($Vvar777_NAME_string) = ('POHNUM')">
        <xsl:value-of select="true()"/>
    </xsl:if>
</xsl:for-each>
name="Vvar775_exists" select="string-length( $ExistsResult22 )&gt;0"/>
test="string($Vvar775_exists)='true' or string($Vvar775_exists)='1'">
    <AlertMessage>
        <xsl:value-of select="concat(concat(concat(concat(concat('Error encountered during explosion for WipOrderNo = ',
$Vvar773_FLD_string), ', WipOrderType = '), '16'), ', JobDescription = '), concat('ADXCOMMANDES - ', $Vvar773_FLD_string))"/>
    </AlertMessage>
select="$Vvar1_firstSource/GRP">
select="FLD">
    <xsl:variable name="Vvar795_FLD_string" select="string(.)"/>
    <xsl:variable name="ExistsResult23">
    <xsl:for-each select="@NAME">
        <xsl:variable name="Vvar799_NAME_string" select="string(.)"/>
        <xsl:if test="($Vvar799_NAME_string) = ('POHNUM')">
            <xsl:value-of select="true()"/>
        </xsl:if>
    </xsl:for-each>
    </xsl:variable>
    <xsl:variable name="Vvar797_exists" select="string-length( $ExistsResult23 )&gt;0"/>
test="string($Vvar797_exists)='true' or string($Vvar797_exists)='1'">
    <xsl:for-each select="$Vvar722_LIN/@NUM">
        <xsl:variable name="Vvar804_NUM_byte" select="number(.)"/>
        <Name>
            <xsl:value-of select="concat($Vvar795_FLD_string, format-number(number($Vvar804_NUM_byte),
'0000'))"/>
        </Name>
    </xsl:for-each>

```

```

<xsl:for-each
    <xsl:for-each select="FLD">
        <xsl:variable
            <xsl:variable
                <xsl:for-
                    </xsl:variable>
                    <xsl:variable
                        <xsl:if
                            </xsl:if>
                            </xsl:for-each>
                            </xsl:for-each>
                            <JobAction>
                            <xsl:for-each
                                <xsl:for-each
                                    </xsl:if>
                                    </xsl:for-each>

```



```

select="FLD">
    <xsl:variable name="Vvar874_FLD_string" select="string(.)"/>
    <xsl:variable name="ExistsResult27">
    <xsl:for-each select="@NAME">
        <xsl:variable name="Vvar878_NAME_string" select="string(.)"/>
        <xsl:if test="($Vvar878_NAME_string) = ('POHNUM')">
            <xsl:value-of select="true()"/>
        </xsl:if>
    </xsl:for-each>
    </xsl:variable>
    <xsl:variable name="Vvar876_exists" select="string-length( $ExistsResult27 )>0"/>
test="string($Vvar876_exists)='true' or string($Vvar876_exists)='1'">
    <Description>
        <xsl:value-of select="concat('ADXCOMMANDES - ', $Vvar874_FLD_string)"/>
    </Description>
    </xsl:if>
    </xsl:for-each>
    </xsl:for-each>
    <Schedule>
    <ExecuteImmediate>
    <xsl:value-of select="string('true')='true' or string('true')='1'">
    </ExecuteImmediate>
    </Schedule>
    </JobSchedule>
    </Job>
    </xsl:if>
    </xsl:for-each>
    </xsl:for-each>
    </JobCreationEvent>
    </OrderHeader>
    </FlexNet_Orders>
    </xsl:template>
</xsl:stylesheet>

```


Titre : PROPOSITION D'UN CADRE DE MODÉLISATION MULTI-ÉCHELLES D'UN SYSTÈME D'INFORMATION EN ENTREPRISE CENTRÉ SUR LE PRODUIT

Résumé : Notre thèse porte sur la Modélisation d'Entreprise et plus particulièrement sur une vision Ingénierie Système de celle-ci afin de faciliter son application « à minima » en entreprise. Alors que la modélisation d'entreprise arrive à maturité scientifique et technologique, force est de constater que sa pratique est encore trop peu courante. En outre, il n'existe pas à proprement parler de tâche de modélisation, de maintenance et de gestion de modèles dans l'entreprise. La question s'est alors posée de la relation d'interopération entre les systèmes, d'un Système-Entreprise évolutif, le plus souvent composé de façon ad-hoc de sous-systèmes (COTS), qui ont pour particularité d'encapsuler leur propre Savoir-Faire et leurs Ingénieries respectives, en incluant le Système-Produit (son besoin, son projet, ...) à la source de chaque recomposition « à la demande » du Système-Entreprise. Un des objectifs de nos travaux est donc de faciliter l'interopération entre les ingénieries de ces systèmes interopérants (autour du produit A Faire) et l'Ingénierie du Système-Projet d'Entreprise (Pour Faire). Nous nous sommes alors inspirés de la proposition de Kuras pour formaliser notre Système-Entreprise (en fait son Ingénierie) dans une sorte de « Système de Systèmes d'Information » (SdSI) en tenant compte de ses aspects récursifs et multi-échelles dans un contexte organisationnel évolutif. De plus, en nous appuyant sur la théorie des patterns, nous avons identifié puis dérivé le « composite-pattern » pour mieux décrire la nature « Tout et Partie » de l'Holon Système-Entreprise utilisé par Kuras pour extraire le patron de conceptualisation le plus adéquat. Parmi tous les points de vue d'un système, cette dérivation nous permet de représenter ceux du Système A Faire et ceux du Système Pour Faire avec son contenu d'Ingénierie associée guidée par le cadre de modélisation de Zachman. Nous avons outillé notre proposition de cadre d'Ingénierie Système Basée sur des Modèles (ISBM) en étendant le méta-modèle de l'outil « MEGA Modelling Suite » ainsi qu'en développant les interfaces utilisateurs nécessaires et nous l'avons appliqué à l'Ingénierie d'un Système de Traçabilité d'un Produit dans le contexte d'un scénario PLM plausible entre l'AIPL et DIMEG.

Mots clefs : modélisation d'entreprise, interopération, intégration, systèmes d'information, système de systèmes, récursivité, multi-échelles, Zachman, ingénierie système, ingénierie basée sur les modèles, pattern.

Title : PROPOSAL OF A MULTI-SCALE MODELLING FRAMEWORK FOR A PRODUCT-CENTRIC ENTERPRISE INFORMATION SYSTEM

Abstract: Today's needs for more capable enterprise systems in a short timeframe are leading more organizations toward the integration of existing component-systems. In the domain of Enterprise Modelling and more generally on Systems Engineering, the key issue is related to aligning enterprise models with the composite structure of the modelled systems. While Enterprise Modelling matures both scientifically and technologically, it is clear that its practice is still uncommon in the enterprises. There is no modelling approach in enterprise, neither even capitalization of models. This paper aims at proposing a Model-Based System Engineering (MBSE) approach, by using the recursion principle, together with a "multiple scales system thinking", for capitalizing the engineering know-how, over time, during a project lifecycle. In this context, we advocate that, within an enterprise, the heterogeneous set of Information Systems may be considered as COTS (Commercial-Off-The-Shelf), encapsulating their own expertise (engineering), and contributing all together to a single System-of-Information System (SoIS). We are then proposing a SoIS engineering methodology resulting from deriving our MBSE approach with a recursive view of the Zachman framework. This approach, based on modelling rules and enterprise modelling constructs, is contributing to ensure the consistency and the completeness of the various heterogeneous models interoperating during systems engineering projects. To validate our MBSE proposal, a modelling tool has been prototyped. It helped at applying our SoIS engineering methodology in a use case scenario related to the requirement analysis and the specification of a product traceability system.

Keywords: Enterprise Integration, Systems Interoperability, Systems Engineering, System-of-Systems, Enterprise Modelling, Information Systems, recursion, Zachman, multi-scales, pattern, Model Based Systems Engineering