



*Usinor Productivité Plus*

*ASC Sud*

UNIVERSITE MONTPELLIER II, SCIENCES ET TECHNIQUES DU LANGUEDOC

**INSTITUT UNIVERSITAIRE PROFESSIONNALISE**

**Génie Mathématiques et Informatique**

**Septembre 2001 / janvier 2002**

# **Dictionnaire EPC : données et modèles**

Fabrice Berna

**Directeurs de Stage :**

Gérard PASQUET, Edgard LOPEZ

**Remerciements**

Je tiens à remercier mon maître de stage, Monsieur **Gérard PASQUET**, chef de projet de l'EPC, et Monsieur **Edgard LOPEZ**, chef de projet Dictionnaire, qui m'ont accueilli dans leur équipe et qui m'ont confié un projet très enrichissant.

J'adresse également mes remerciements à mon maître de stage technique, Monsieur **Jérôme GUY**, avec qui j'ai travaillé. Malgré son travail important, il a su trouver la disponibilité nécessaire pour me conseiller et me guider dans mon projet.

Je tiens aussi à remercier Messieurs **Hung NGUYEN et Marc MICHAUD**, ainsi que toutes les personnes de l'équipe EPC pour leur accueil chaleureux et leurs conseils avisés. Ils m'ont permis d'effectuer mon stage dans de bonnes conditions.

**Titre du rapport :**

Dictionnaire EPC, données et modèles

**Résumé :**

Le stage s'est déroulé à Usinor SI, service informatique du Groupe Usinor, premier producteur d'acier européen.

Le stage s'inscrit dans le cadre du projet EPC (Evaluation Prévisionnelle des Capabilités), projet permettant de qualifier les produits au rythme de la production, de détecter les écarts prévu/réalisé, et d'effectuer des diagnostics sur ces informations.

Durant le stage, il s'agit de développer une application Dictionnaire, véritable livre ouvert sur la configuration de l'EPC. Le Dictionnaire permettra d'afficher l'ensemble des données manipulées par l'EPC, les liens entre ces données, et les traitements effectués sur ces données.

Le Dictionnaire est une application intranet évoluée, qui traite des fichiers au format XML. Elle n'envoie au client que du code HTML, c'est à dire que tous les traitements de présentation s'effectuent côté du serveur et non côté client.

**Mots Clés :**

Application Intranet, Java, XML, XSL, XSLT, Java côté serveur, Struts, Framework MVC2, JSP, Servlet

**Maître de stage :**

Gérard PASQUET, Chef de projet EPC, Usinor SI, ASC Sud.

**Durée du stage :**

Quatre mois, du 17 septembre 2001 au 18 janvier 2002

<b>REMERCIEMENTS</b> .....	<b>1</b>
<b>RESUME</b> .....	<b>2</b>
<b>SOMMAIRE</b> .....	<b>3</b>
<b>TABLE DES FIGURES</b> .....	<b>5</b>
<b>INTRODUCTION</b> .....	<b>5</b>
<b>LE GROUPE USINOR</b> .....	<b>6</b>
PRESENTATION .....	7
STRATEGIE D’USINOR .....	7
CHIFFRES CLES 2000 .....	8
LE PROJET ARCELOR .....	9
LE SERVICE USINOR SI .....	9
<b>CADRE DU STAGE</b> .....	<b>10</b>
LE PROJET EPC .....	10
MODULE 2 DE L’EPC .....	11
EXEMPLE D’UTILISATION DE L’EPC .....	12
<b>LE PROJET DICTIONNAIRE</b> .....	<b>14</b>
PRESENTATION .....	14
OBJECTIFS .....	15
PRINCIPE DU DICTIONNAIRE .....	16
FICHIERS MANIPULES .....	18
<b>PLANNING D’ACTIVITE</b> .....	<b>19</b>
INTEGRATION A L’EQUIPE DU DICTIONNAIRE .....	19
ETUDE .....	20
REALISATION DU PROJET .....	20
DOCUMENTATION.....	21
FRONTAL.....	21
<b>TECHNOLOGIES MISES EN ŒUVRE</b> .....	<b>23</b>
ARCHITECTURE CIBLE .....	23
XML .....	24
XSL .....	24
MANIPULATION DU XML .....	24
DOM.....	25
JAVASCRIPT .....	25
CSS.....	25
JSP.....	25
LE MODELE-VUE-CONTROLEUR .....	26
<b>CONCEPTION DU DICTIONNAIRE</b> .....	<b>29</b>
PRINCIPE DU DICTIONNAIRE EPC SOUS STRUTS .....	29
COMMUNICATION ENTRE LES COMPOSANTS .....	31

CONFIGURATION DE L'APPLICATION DICTIONNAIRE .....	32
<b>APPLICATION DE CONTROLE.....</b>	<b>34</b>
PRINCIPE .....	34
LES DIFFERENTES CLASSES .....	34
FICHIER RESULTAT .....	35
<b>LES APPORTS DU STAGE.....</b>	<b>36</b>
POUR USINOR SI .....	36
POUR LE STAGIAIRE .....	36
<b>CONCLUSION .....</b>	<b>38</b>

## Table des Figures

Figure 1 Chiffre d'affaires par zone.....	8
Figure 2 Répartition des productions .....	8
Figure 3 Répartition des salariés par pays.....	9
Figure 4 Modules de l'EPC.....	11
Figure 5 Application EPC sur Laminoir.....	13
Figure 6 Le dictionnaire présente la configuration de l'application EPC.....	14
Figure 7 Le concept AVEM .....	16
Figure 8 Transformation de l'information dans l'EPC M2 .....	18
Figure 9 Planning d'activité .....	19
Figure 10 Le client est léger, le serveur ne renvoie que du HTM.....	23
Figure 11 Cheminement d'une requête .....	26
Figure 12 Architecture de type MVC2.....	27
Figure 13 Fonctionnement de Struts.....	28
Figure 14 Principe du dictionnaire avec une architecture Struts.....	29

## Introduction

Le stage de quatre mois s'inscrit dans le cursus de la dernière année de formation en IUP Génie Mathématiques et Informatique. Il s'est déroulé au sein de la société USINOR SI, service informatique du groupe USINOR, premier producteur européen d'acier.

Le stage s'est effectué dans le cadre d'un vaste projet informatique de développement d'une application dans le domaine sidérurgique. Cette application, l'EPC (Evaluation Prévisionnelle des Capabilités), permet de qualifier les produits au rythme de la production, détecter les écarts entre les prévisions et les mesures effectuées, et simuler la maîtrise statistique des outils de production. L'EPC est encore en développement et entre dans sa phase d'industrialisation (développement du produit final qui sera mis en production).

L'application EPC est paramétrée par un ensemble complexe de fichiers au format XML. Le problème actuel est qu'il n'existe aucune méthode autre qu'un éditeur de texte pour consulter et visualiser le contenu de ces fichiers ou contrôler leur cohérence, ce qui rend la tâche de configuration assez complexe.

Le but du stage sera de satisfaire à ce besoin en créant une application Dictionnaire, véritable livre ouvert sur la configuration, sur les données manipulées et sur les traitements effectués par l'application EPC.

J'ai choisi ce stage parce que le projet proposé me donnait l'occasion d'approfondir mes connaissances en Java, en XML et XSL, et en technologies Intranet. De plus, ce stage était pour moi l'occasion de découvrir et participer au développement d'un projet informatique appliqué à un processus industriel sidérurgique, dans un grand groupe d'envergure internationale.

Je vais dans un premier temps présenter le Groupe USINOR, et son Service Informatique Usinor SI. Je me consacrerai dans un deuxième temps sur le déroulement du stage avec un aperçu du travail effectué, des problèmes rencontrés et des solutions apportées.

Je terminerai enfin par présenter ce que m'a apporté ce stage en dehors du cadre de ma mission.

## **Présentation**

---

Avec un chiffre d'affaires de 15,7 milliards d'euros, et une production de 21 millions de tonnes en 2000, USINOR, groupe industriel privé coté en bourse, est le premier producteur d'acier d'Europe, et figure parmi les premiers mondiaux.

USINOR est spécialisé dans la production, la transformation et la distribution d'aciers inoxydables et d'aciers plats au carbone destinés principalement aux marchés de l'automobile, de la construction, de l'emballage, et de l'électroménager.

Ainsi en Europe, les aciers USINOR

- √ forment la caisse et la carrosserie d'une voiture sur trois
- √ composent les murs et la toiture d'un bâtiment sur cinq
- √ servent à fabriquer une boîte d'emballage (boîtes de boisson, conserves alimentaires, emballages divers) sur trois
- √ représentent un tiers des aciers des appareils électroménagers (froid, lavage, cuisson).

Le Groupe accompagne le développement de ses clients partout dans le monde en élaborant avec eux des « solutions aciers » adaptées à leurs besoins.

L'acier est un produit bon marché qui a l'avantage sur bien d'autres matériaux d'être recyclable à l'infini, et d'être facilement trié dans les opérations de recyclage puisqu'il est magnétique. USINOR réalise quarante pour cent de sa production par recyclage d'aciers usagés.

## **Stratégie d'USINOR**

---

### **La stratégie internationale d'Usinor**

En s'appuyant sur ses positions solidement établies en Europe et en Amérique du Nord, le groupe s'est développé progressivement en renforçant son aval et ses circuits de distribution en Europe. Depuis 10 ans, le groupe a acquis des sociétés ou a créé, grâce à des partenariats, de nouveaux sites de production.

### **En Europe**

L'acquisition d'Arvedi et de La Magona en Italie et du pôle sidérurgique de Sagonte en Espagne a conforté le site méditerranéen de Fos sur Mer tandis que les positions au nord et à

l'est étaient confortées par l'acquisition de Cockerill Sambre en Belgique et d'Eko Stahl en Allemagne.

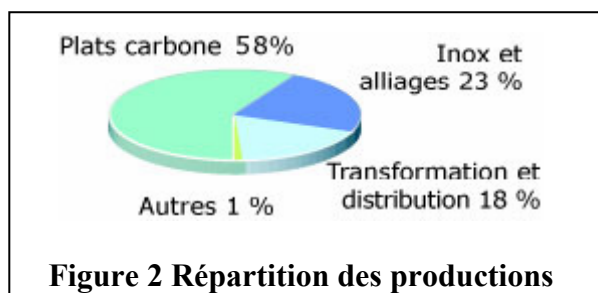
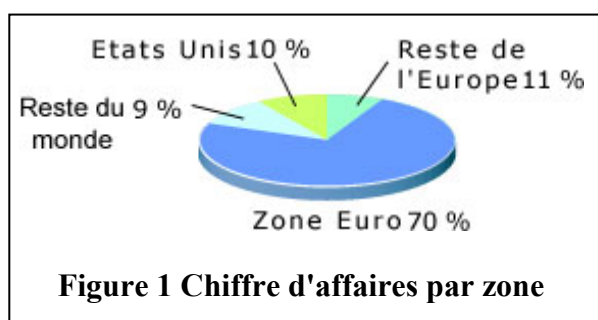
### Hors Europe

Le groupe a également parachevé sa mondialisation en contrôlant J&L (Etats Unis) et Thainox (Thaïlande), en prenant des participations dans Acesita et CST au Brésil et en créant une joint-venture au Canada pour la fourniture de tôles galvanisées à l'automobile.

### Chiffres clés 2000

L'acier est l'un des matériaux les plus utilisés dans le monde. Son usage est en développement constant.

√ **Chiffre d'affaires** : 15,7 milliards d'euros

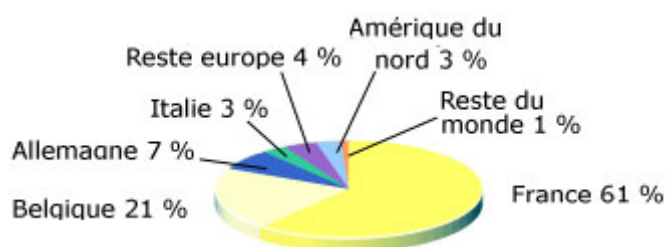


√ **Bénéfice d'exploitation** : 1,09 milliards d'euros

√ **Dividende net par action** : 0,56 euros

√ **Production d'acier brut** : 21 millions de tonnes. Usinor est ainsi le premier producteur européen en tonnage et le 5<sup>ème</sup> mondial.

√ **Effectifs mondiaux** : 62.000 personnes



**Figure 3 Répartition des salariés par pays**

### Le projet Arcelor

ARBED, ACERALIA et USINOR ont signé, le 19 février 2001, un document d'intention exprimant le projet de réunir au sein d'une société nouvelle l'ensemble de leurs forces industrielles, commerciales et humaines. Le nouveau Groupe devrait être créé début 2002. Cette union placera le Groupe au premier rang mondial des producteurs d'acier.

#### **Quelques chiffres...**

- √ **Chiffre d'affaires** : 30 milliards d'euros
- √ **Production d'acier brut** : 46 millions de tonnes
- √ **Effectifs** : 110.000 personnes

ACERALIA, ARBED et USINOR constitueront un groupe multi-métiers capable d'accompagner ses clients sur l'ensemble de leurs marchés. Leur ambition commune s'exercera en tirant le meilleur parti des atouts dont ils disposent aujourd'hui dans chacun de leurs secteurs : aciers plats au carbone, aciers longs au carbone, aciers inoxydables, distribution, et transformation, où ils sont dans chaque cas leader ou co-leader mondial.

### USINOR SI

Usinor SI est la société de services en Systèmes d'Information du Groupe Usinor. Sa mission est de développer, intégrer et maintenir les systèmes d'information du Groupe. Le niveau de qualité de ses prestations est d'une grande importance stratégique pour Usinor. Son capital

intellectuel alliant la connaissance des métiers du Groupe, celle de la technologie informatique et de la méthodologie constituent son patrimoine essentiel.

Pour des raisons de clarté et d'efficacité de fonctionnement, Usinor SI s'est dotée d'une organisation de type Société commerciale indépendante.

**Cadre du Stage**

## **Le projet EPC**

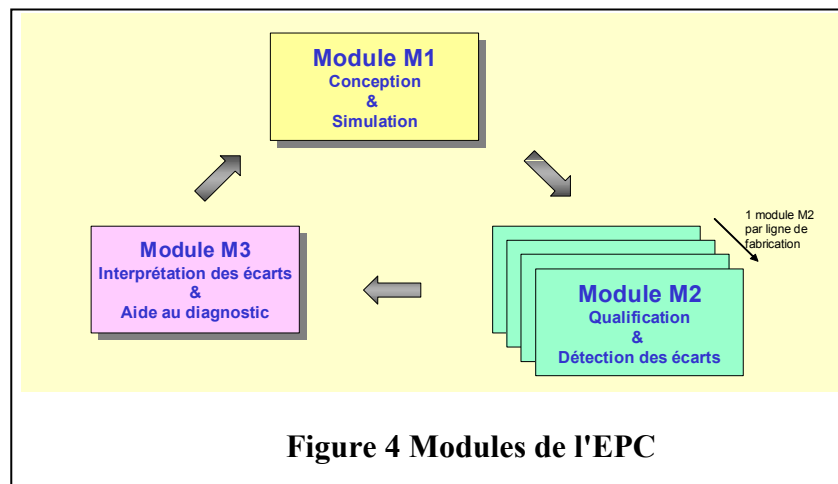
---

Le projet EPC (Evaluation Prévisionnelle des Capabilités) est un grand projet informatique de développement d'une application permettant de :

- √ qualifier les produits (Brames, Coils, Bobines) en temps réel (au rythme de la production)
- √ concevoir, tester et mettre en production de nouveaux modèles et d'aider aux diagnostics d'écart entre les mesures et les résultats de modèles
- √ simuler la capacité (maîtrise statistique) des outils de fabrication (Evaluation Prévisionnelle des Capabilités)

Il est composé de trois modules :

- √ le module M1 : module de conception et simulation
- √ le module M2 : module de repérage des écarts prévu/réalisé. Ce module est opérationnel et en prise directe sur les lignes de production.
- √ le module M3 : interprétation des écarts et aide au diagnostic



## **Module 2 de l'EPC**

L'objectif du module 2 de l'EPC est de fournir au métallurgiste un outil permettant de :

- √ suivre et caractériser en quasi-temps réel les produits fabriqués sur la ligne

- √ vérifier rapidement et de façon synoptique la stabilité des conditions d'exploitation de la ligne
- √ être alerté très vite sur la dérive d'un capteur ou sur une modification des populations traitées
- √ conduire simplement des enquêtes et des opérations de diagnostic.

Chaque ligne dispose de « son » EPC-M2.

Les enjeux pour la fonction « Fabrication et Contrôle » sont les suivants :

- √ Contrôler la production en temps réel afin notamment d'éviter les déclassements
- √ Contrôler la stabilité de la ligne
- √ Contrôler la validité des mesures process (mesures reçues des capteurs) et des caractéristiques produits.

### **Exemple d'utilisation de l'EPC**

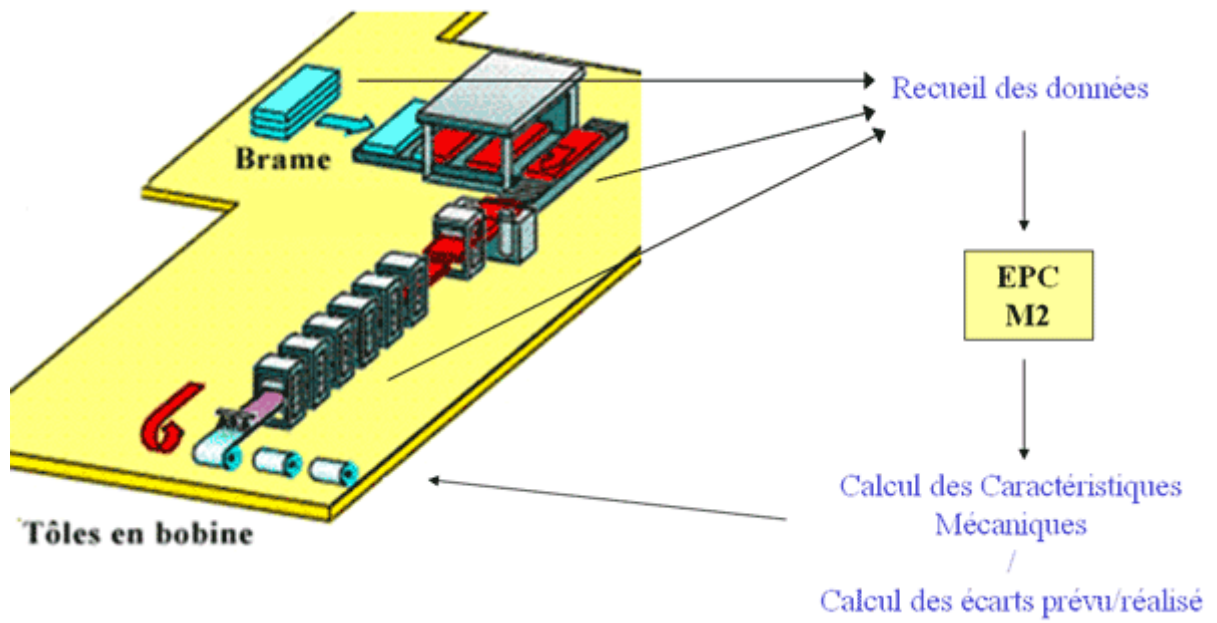
---

Actuellement, Pour définir les caractéristiques mécaniques d'un produit, on effectue le prélèvement d'un échantillon, que l'on envoie en laboratoire, sur une portion du produit en bout de chaîne.

L'objectif de l'EPC est d'effectuer un recueil permanent des informations tout au long de la chaîne de production pour prévoir entre autre quelles seront les caractéristiques de ce produit.

Les avantages d'un tel procédé sont :

- √ Une qualification des produits en temps réel
- √ Le calcul des caractéristiques mécaniques sur toute la longueur du produit et non plus sur une seule portion



**Figure 5 Application EPC sur Laminoir**

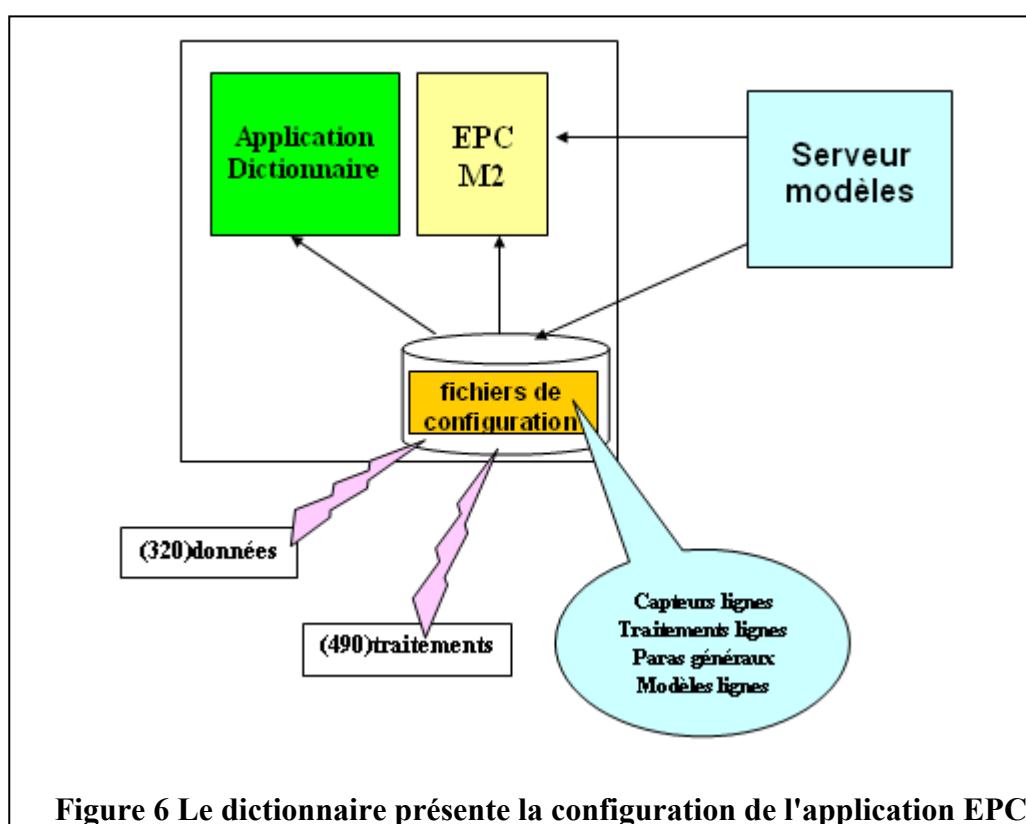
## Le projet Dictionnaire

### Présentation

L'application dictionnaire permet de visualiser les traitements effectués sur les données de ligne dans l'EPC. Elle est le reflet de la configuration de l'EPC pour une ligne donnée.

Le paramétrage de l'application EPC M2 se présente sous forme de fichiers au format XML. Ces fichiers se situent dans un répertoire propre à la ligne. L'application Dictionnaire doit lire ces fichiers de configuration et présenter les données, les liens entre celles-ci, et tous les traitements qui leur sont appliqués.

Le Dictionnaire est un véritable « livre ouvert » sur l'application EPC. Il récupère et visualise les informations directement sur chaque serveur EPC M2.



## Objectifs

---

Les fonctions attendues dans le cadre du projet Dictionnaire sont :

- √ Consulter via une interface intranet :
  - La description de la ligne de production
  - L'ensemble des données brutes et élaborées, ainsi que leurs caractéristiques détaillées, utilisées par l'application EPC M2 sur chaque ligne. Ce sont : les données d'entrée, les capteurs, les modèles, les données élaborées, et les données de sorties.
  - Les opérations de transformation de ces données
  - L'enchaînement complet de ces opérations, en décrivant chaque niveau d'élaboration du concept AVEM (cf. paragraphe suivant).
- √ Outil d'administration permettant de vérifier la cohérence et l'intégrité des fichiers de paramètre d'une ligne de l'EPC M2.
- √ Pouvoir rechercher une donnée dans la base d'informations. La recherche s'effectuera en saisissant une chaîne de caractère et en précisant sur quel type et quels champs de la donnée la recherche doit s'appliquer.
- √ Une fonction de tri des données. Lorsque les informations sont affichées en liste, il devra être possible de les trier par ordre alphabétique sur un champ précis.
- √ Pouvoir accéder directement au détail d'une donnée par construction d'une requête
- √ Permettre de paramétrer l'application :
  - Les informations devront être présentées sous une forme multilingue. Ces informations seront stockées dans un fichier contenant tous les libellés utilisés dans l'application : traduction.xml
  - La liste des serveurs et lignes consultables est contenue dans le fichier serveursEPC.xml. Une interface d'administration permettra de modifier ces informations.
  - Un fichier propre à l'application : dico-config.xml, permet de paramétrer l'application sans intervenir sur le code. Il contient notamment comme information le chemin d'accès aux fichiers XSL, le chemin d'accès à la liste des modèles ainsi que les fichiers de description de modèle, et d'autres informations utiles.

## Principe du Dictionnaire

### Le concept AVEM

L'application EPC M2 effectue en temps réel un recueil d'informations reçues de la chaîne de production. L'analyse de l'information se fait en quatre étapes :

- ✓ **Acquisition** : réception de l'information à l'état brut
- ✓ **Validation** : Traitement de validation appliqué sur la donnée brute pour obtenir une information appelée capteur
- ✓ **Elaboration** : Traitements appliqués sur des informations capteur pour obtenir des données d'entrée de modèle, appelées données élaborées
- ✓ **Modélisation** : Les modèles reçoivent en entrée les données élaborées. Ils effectuent des calculs et renvoient en sortie des informations appelées données de sorties

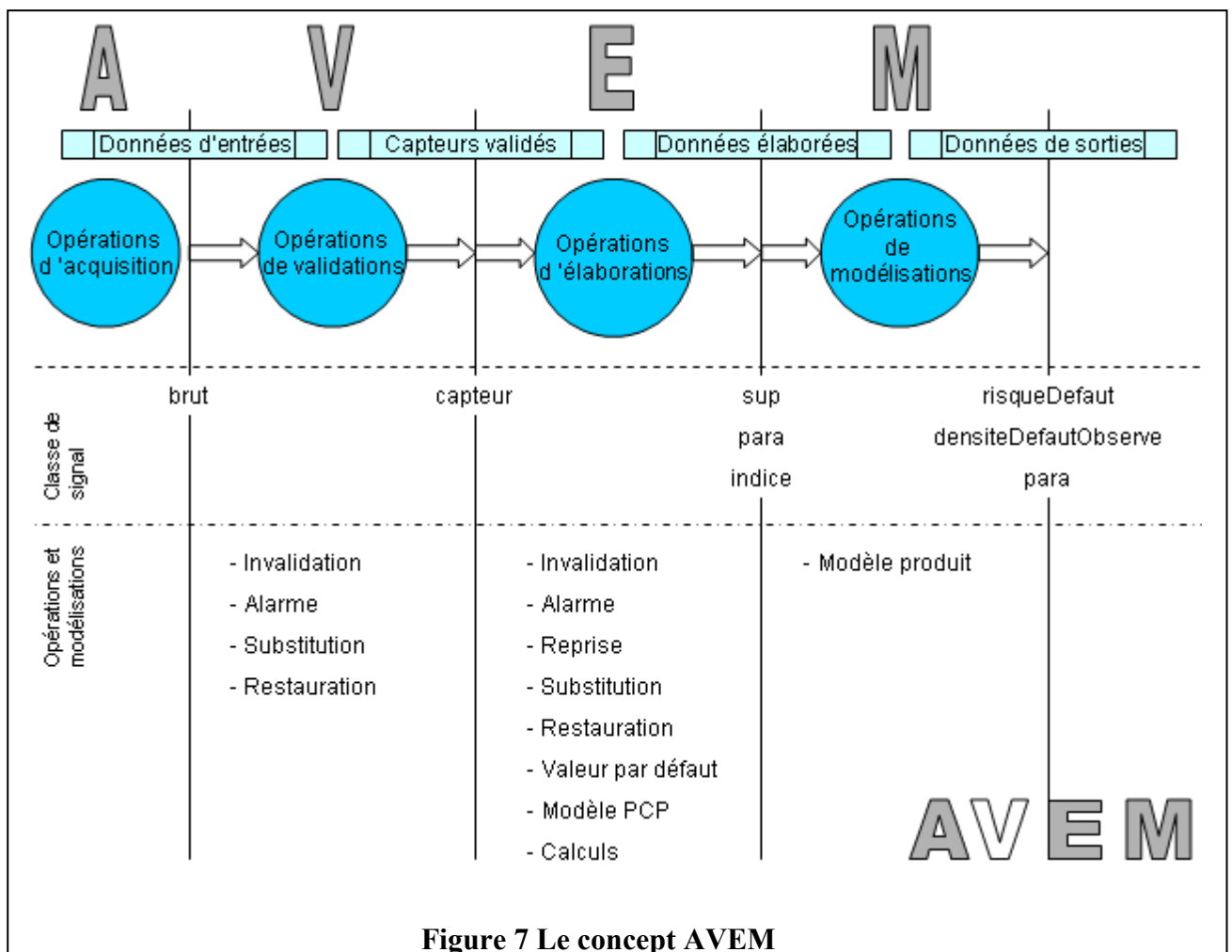


Figure 7 Le concept AVEM

## Description des données manipulées par le Dictionnaire

### Données de description de ligne

Ce sont les données de description d'une ligne de production. Une ligne est un maillon du processus sidérurgique. Elle transforme un produit brut ou semi-fini en un produit semi-fini ou fini. Par exemple, le haut fourneau reçoit en entrée du charbon et du minerais de fer, et transforme le tout en fonte.

Actuellement, l'EPC est utilisée sur trois types de ligne : les aciéries, les trains à chaud, et la galvanisation.

D'un point de vue EPC, les données de description de ligne contiennent le détail des sources d'information, les types de message reçus, le paramétrage de la ligne, et des informations techniques diverses.

### Données d'entrée

Ce sont toutes les informations qui arrivent à l'état brut dans l'application EPC. Ces données proviennent généralement de l'informatique de process ou l'informatique de gestion.

### Capteurs validés

Les capteurs sont obtenus par validation de données d'entrées. Ils ont un sens métier, et sont manipulés dans l'application EPC.

### Données élaborées

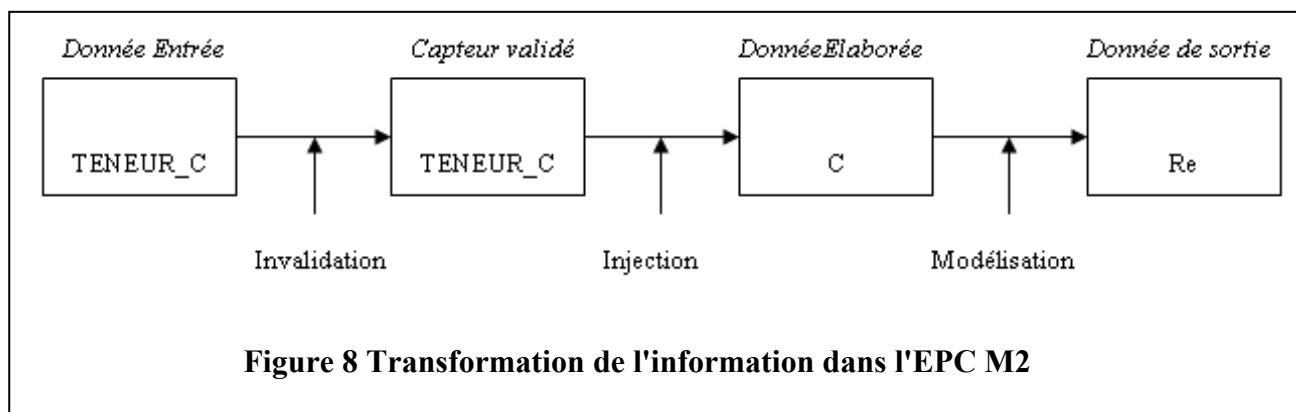
Ce sont des données élaborées à partir de l'ensemble des signaux disponibles. Elles sont parfois nécessaires à la construction d'élaborations ainsi qu'à l'obtention de paramètres.

### Modèles

Les modèles sont des calculateurs créés par des mathématiciens. On distingue les modèles CM (Caractéristiques Mécaniques), PCP (Paramètres Critiques Process), ou défauts. Chaque modèle permet de calculer, selon sa fonction, les caractéristiques prévues pour le produit, ou de calculer les écarts entre les caractéristiques mécaniques calculées et celles mesurées.

## Sorties

Ce sont les résultats calculés par les modèles.



## Fichiers manipulés

Les fichiers de configuration de l'EPC sont des fichiers au format XML. L'ensemble des fichiers manipulés par le Dictionnaire est le suivant :

- √ Capteurs\_nomLigne.SMArt : Informations relatives aux capteurs utilisés
- √ DonneesEntrees\_nomLigne.SMArt. Ce fichier contient des informations sur tous les types de données : provenance, flux, section et type
- √ InfosDonnees\_nomLigne.SMArt . Ce fichier est complémentaire au fichier DonneesEntrees : libellé, description, documentation et catégorie.
- √ Ligne\_nomLigne.SMArt et DescriptionLigne\_nomLigne.SMArt : ces fichiers contiennent toutes les informations relatives à la description de la ligne
- √ ModelesCM\_nomLigne.SMArt. Ce fichier contient la liste des modèles de Caractéristiques mécaniques qui sont utilisés sur cette ligne.
- √ traitements\_nomLigne.SMArt. Ce fichier est le plus important. Il contient toutes les informations relatives aux traitements effectués par l'application EPC.
- √ InventaireModele.SMArt. Ce fichier contient l'inventaire de tous les modèles utilisables ainsi que le chemin d'accès au fichier de ce dernier.
- √ Ensemble des fichiers contenant toutes les informations ayant trait aux modèles

## Planning d'Activité

	Septembre				Octobre				Novembre				Décembre				Janvier					
Semaine	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5
Intégration																						
Etude																						
Réalisation																						
Documentation																						
Frontal																						
Rapport de Stage																						
Soutenance Stage																						

Figure 9 Planning d'activité

### Intégration à l'équipe du Dictionnaire

#### Documentation :

Documentations diverses sur : le Groupe Usinor, l'Usine Sollac, le service Usinor SI, le Projet EPC, et le Dictionnaire EPC.

Formation aux technologies qui seront utilisées : XML, XSL, XSLT, Java coté serveur (JSP, Servlets, JavaBeans).

#### Introduction au projet Dictionnaire :

L'introduction au projet Dictionnaire s'est faite par le développement de quelques fonctionnalités dans la maquette du Dictionnaire :

- √ Page d'accueil multilingue
- √ Affichage de la liste des paramètres dans l'onglet « Données Elaborées »
- √ Création de l'onglet « Documentation »
  - Affichage du Concept AVEM
  - Documentation Cèdres (déploiement, métallurgie)
  - Liens utiles

## **Etude**

---

Après intégration à l'équipe du Dictionnaire, on m'a confié le développement de l'application Dictionnaire finale, en intégrant la migration de l'application vers une nouvelle technologie : Struts.

L'étude s'est faite principalement sur l'aspect architectural :

- √ Etude de de la méthode MVC2
- √ Etude de l'environnement Struts
- √ Test d'interfaces d'aide au développement sous Struts
- √ Conception de l'architecture du Dictionnaire sous Struts

## **Réalisation du projet**

---

La réalisation propre de l'application a pris la majeure partie de la durée du stage.

Elle a constitué en :

- √ Organisation des classes
- √ Configuration de Struts pour le Dictionnaire
- √ Passage du dictionnaire à la technologie Struts
- √ Gestion des messages d'erreur

## **Publication**

La publication sur le serveur WebSphere a posé quelques problèmes. En effet Struts est une technologie très récente, et le serveur WebSphere n'était pas optimisé pour ce type d'application. Il a fallu procéder à un ensemble de modifications : patch pour WebSphere, et modification du code source de Struts.

## **Gestion des accès concurrents**

Une des spécificités de Struts est de n'instancier qu'une seule fois chaque classe. Ceci implique que toutes les variables d'instance sont partagés par tous les utilisateurs. La gestion des accès concurrents s'en trouve modifiée : il ne faut déclarer que des variables locales (dans

les méthodes). Il a donc été nécessaire de modifier toute la structure des classes de l'application Dictionnaire.

### **Fonctions spécifiques**

Des fonctions spécifiques au Dictionnaire ont ensuite été développées. Il s'agit de :

- √ Accès direct à une donnée
- √ Optimisation des algorithmes de calcul
- √ Méthode de recherche
- √ Fonctions de tri
- √ Formulaire de saisie/modification des données serveurs
- √ Fonctions de contrôle

### **Documentation**

---

La documentation est un élément essentiel à USINOR SI. Tout travail nécessite un retour. Ainsi, durant mon stage, j'ai été amené à rédiger les documentations suivantes :

- √ Struts et la méthode MVC2
- √ Comment ajouter Struts à une application web sur WebSphere 3.5
- √ Tests d'interface d'aide au développement sous Struts
- √ Réalisation technique du Dictionnaire
- √ Documentations de Conception pour l'onglet 320 et l'affichage des indices.
- √ Algorithmes de calcul utilisés dans le Dictionnaire
- √ Article de présentation de l'application Dictionnaire

### **Frontal**

---

Le frontal est le premier site auquel on accède avant de consulter le dictionnaire. Il s'agit en quelque sorte de la « couverture » du Dictionnaire.

Il propose :

- √ La présentation du Dictionnaire
- √ Un guide Utilisateur

- √ Les Schémas des fichiers XML
- √ La JavaDoc
- √ La CyberDoc (accès à la documentation du Dictionnaire publiée sur un logiciel interne de gestion des documentations)
- √ Les documentations techniques liés au Dictionnaire
- √ Une documentation Technique Struts

## Technologies mises en œuvre

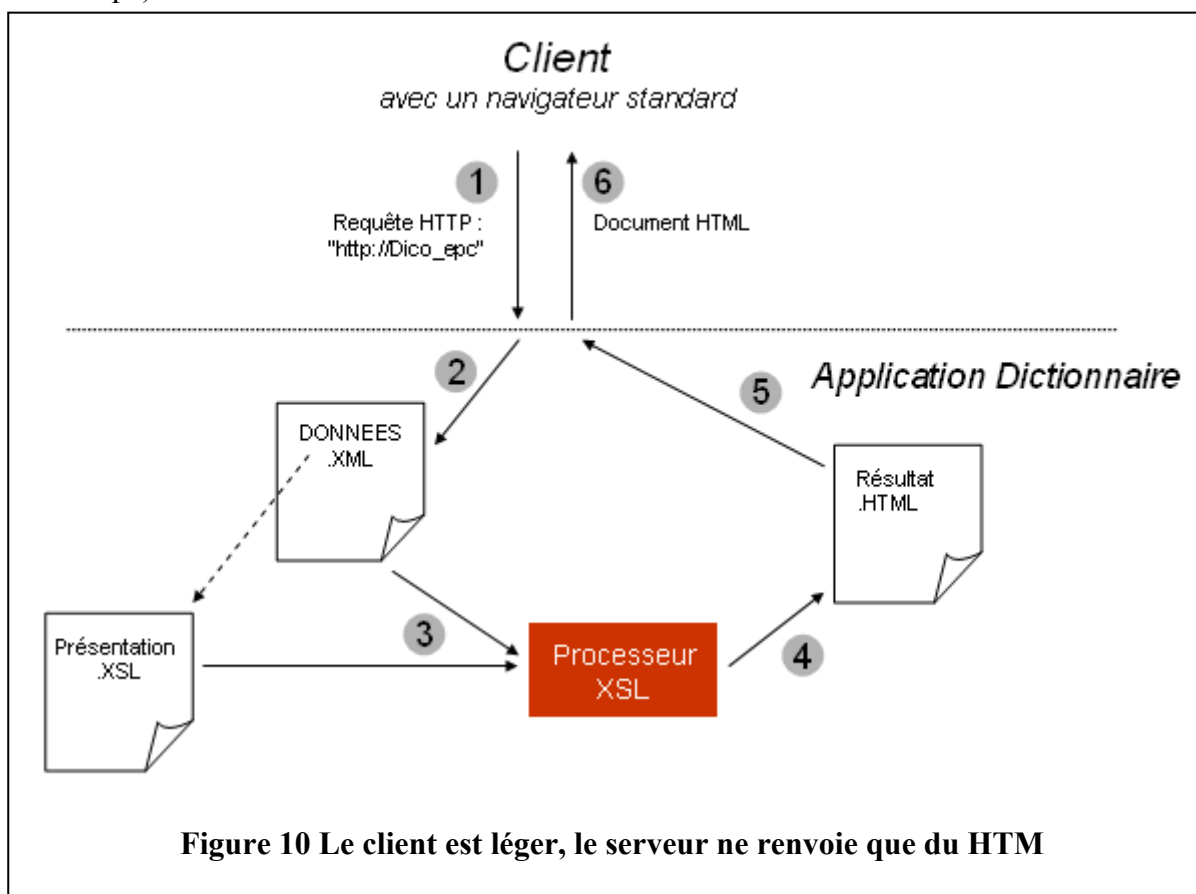
### Architecture cible

L'application dictionnaire est une application intranet qui s'exécutera sur un serveur d'application WebSphere. Elle sera accessible via un navigateur web standard.

Etant donnée le parc informatique (utilisation de Internet Explorer 4), les clients sont des clients dits légers. C'est à dire qu'aucun traitement n'est fait au niveau du navigateur. Contrairement à un client de type évolué qui intègre un processeur XSL, le client léger ne peut effectuer un affichage à partir d'un fichier XML et son fichier XSL associé.

Il faudra donc que tous les traitements complexes soient effectués au niveau du serveur (serveur Evolué). La transformation HTML s'effectuera sur le serveur. Seul un flux HTML transitera entre le serveur et le client.

Ce choix permet aussi d'assurer le bon fonctionnement si d'autres navigateurs sont utilisés (ex : Netscape).



## XML

---

XML : eXtensible Markup Language.

Le XML est totalement ouvert, ce qui rend possible la création de ses propres variables, ses propres balises.

XML est une norme englobante qui peut s'appliquer à de nombreux domaines d'utilisation comme par exemple les mathématiques (MathML), l'image (SVG), et le domaine de la sidérurgie.

L'ensemble des fichiers de configuration de l'EPC utilisent cette technologie. Les fichiers de configuration du Dictionnaire : serveursEPC, traduction, et dicoepc-config seront au format XML pour avoir un support de données uniforme pour toute l'application.

Actuellement, les seuls outils utilisés par le Groupe USINOR pour manipuler les fichiers XML sont des éditeurs basiques de texte, ainsi que des éditeurs spécialisés comme TurboXML et XMLSpy.

## XSL

---

XSL : eXtensible Style Language.

Un fichier XSL est une feuille de styles, dédiée au XML. Il permet de formater un document XML en un document HTML.

Le XSL est utilisé dans le dictionnaire car toutes les données manipulées sont au format XML. XSL est le moyen le mieux adapté pour effectuer le formatage de ces données.

## Manipulation du XML

---

Pour manipuler les fichiers de données XML, on utilisera les technologies Java récentes Xalan et Xerces. Ces deux modules développés par Apache, permettent de :

- √ manipuler , parser (lire), et parcourir les fichiers XML
- √ transformer des données XML et le fichier XSL associé en code HTML

Ces deux modules s'appuient sur le concept DOM (voir ci-après).

## **DOM**

---

DOM : Document Object Model

DOM est une spécification définissant une interface indépendante de toute plateforme et de tout langage, permettant à des programmes et des scripts d'accéder et de mettre à jour dynamiquement le contenu, la structure, et le style des documents. DOM fournit un ensemble standard d'objets pour représenter les documents HTML et XML, et une interface standard pour y accéder et les manipuler.

## **JavaScript**

---

Une partie des événements du Dictionnaire sont gérés à l'aide de JavaScripts. Il s'agit pour le principal de :

- √ Permettre à l'utilisateur d'afficher ou masquer certaines informations lors de la consultation d'une donnée. L'utilisateur peut développer ou non certaines informations (à la manière d'un explorateur windows) et ainsi afficher le niveau de détail qu'il désire.

## **CSS**

---

CSS : Cascading Style Sheets.

Les CSS permettent de contrôler parfaitement la mise en page des différents éléments qui composent le document HTML.

Modifier les CSS permet de modifier l'aspect global du Dictionnaire.

## **JSP**

---

Les pages JSP sont utilisées pour faire une partie de la présentation visuelle du site : elles mettent en forme les parties statiques du Dictionnaire, et s'occupent d'insérer les informations formatées par la partie modèle (processeur XSL).

## Le Modèle-Vue-Contrôleur

Le paradigme de conception MVC (Modèle-Vue-Contrôleur) consiste à organiser l'application en trois composants majeurs : un modèle qui correspond à la logique applicative, une vue qui correspond à la présentation visuelle de l'application et un contrôleur qui définit l'état de la vue en fonction des données gérées par le modèle.

### Les avantages d'une telle décomposition

Cette architecture rend les trois parties de l'application : la logique de présentation, la logique métier et la structure du site web parfaitement indépendantes. Ainsi, la programmation de chacune peut être distribuée à des équipes différentes.

La maintenance de l'application est plus souple. En effet, le partitionnement de l'application permet de modifier la présentation des pages web, sans toucher ni à la structure du site, ni à la logique métier. De même, les composants implémentant la logique métier peuvent être remplacés et la structure du site modifiée, sans toucher à la présentation des pages.

### Traitement d'une requête par le modèle MVC

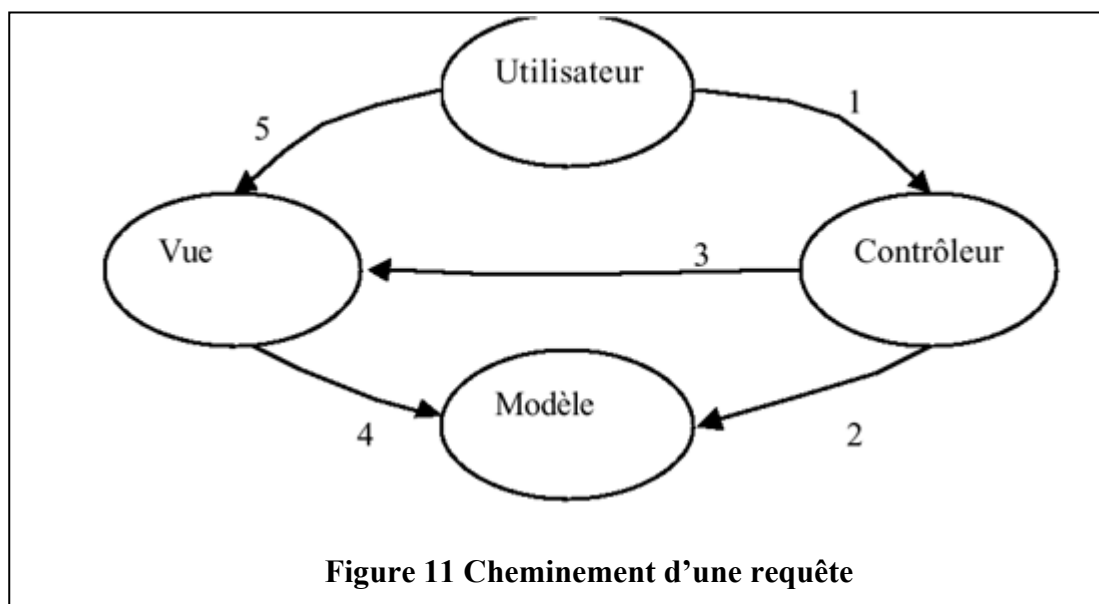


Figure 11 Cheminement d'une requête

1. L'utilisateur manipule l'interface homme/machine. Un événement est envoyé. Cet événement est récupéré par le contrôleur.

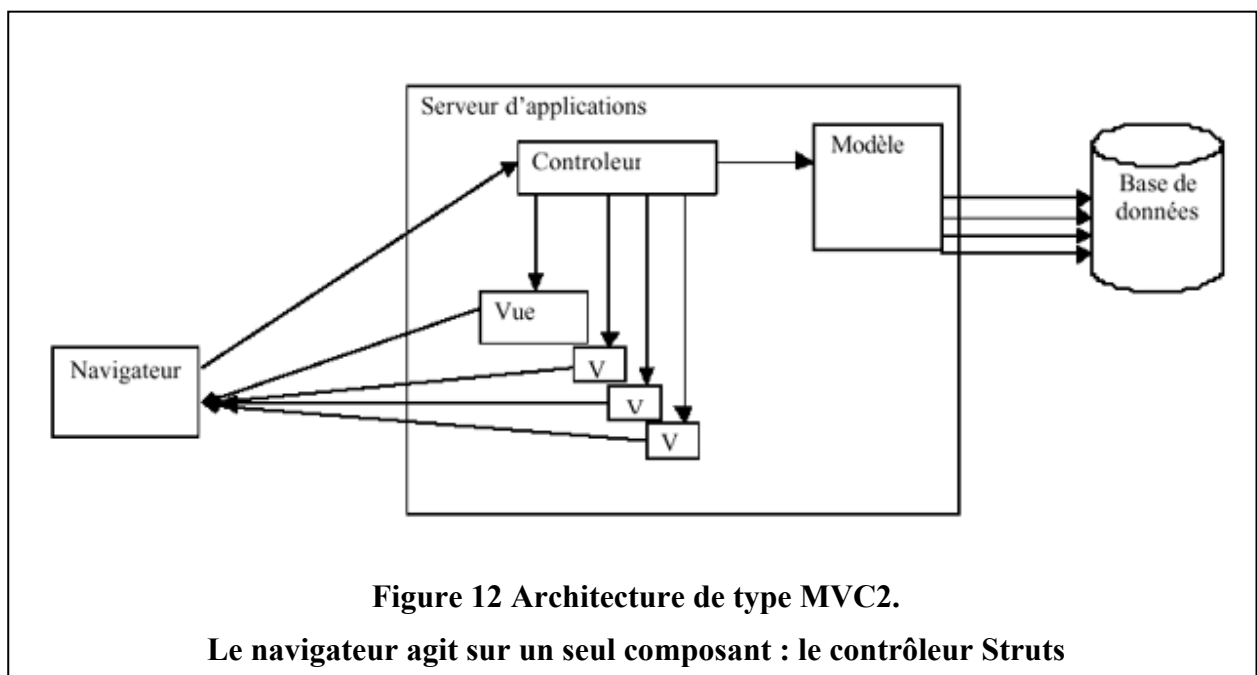
2. Le contrôleur effectue l'action demandée par l'utilisateur en appelant les méthodes nécessaires sur le modèle
3. Le contrôleur informe la vue d'un changement d'état du modèle
4. La vue interroge le modèle afin de connaître son état
5. L'utilisateur voit le résultat de son action

## MVC2

La construction d'un système MVC2 est complexe. Toutes les pages et les composants doivent être inscrits dans le framework, ce qui produit un surcoût de travail.

Chaque page HTML ne peut plus être conçue indépendamment des autres. Elle fait partie d'un processus plus large, contenant :

- ✓ **le modèle** : deux types de JavaBeans pour la logique métier (Action) et pour l'état de l'application (FormBean)
- ✓ **la vue** : des pages JSP qui fournissent au client la réponse à ses requêtes
- ✓ **le contrôleur** : un servlet qui gère les transactions. Il exécute le code nécessaire à une requête, et en coordonne la réponse



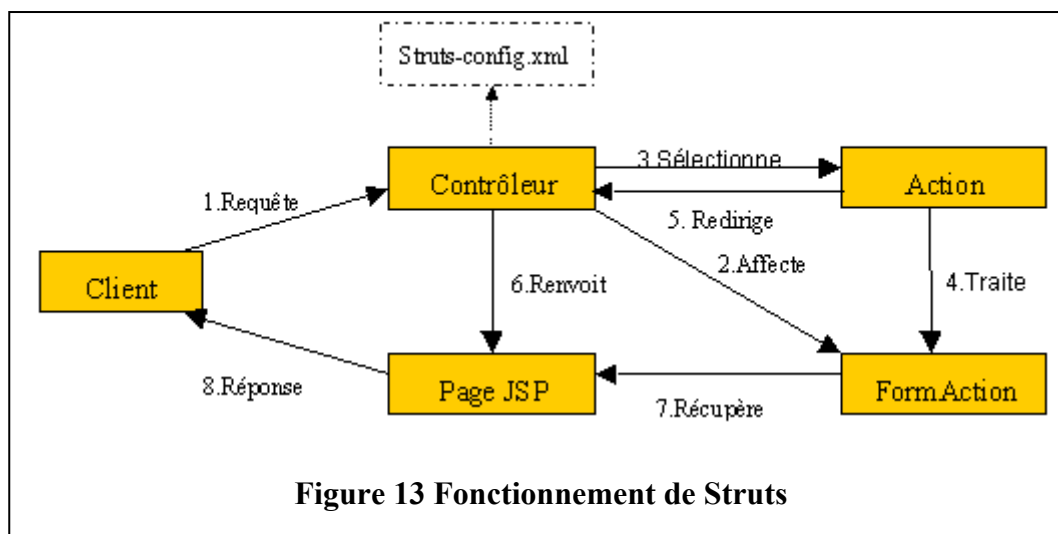
**Figure 12 Architecture de type MVC2.**

**Le navigateur agit sur un seul composant : le contrôleur Struts**

Struts est un Framework de développement d'application web, basé sur le pattern de conception Modèle-Vue-Contrôleur (MVC). Il est composé d'un contrôleur (un servlet), de la vue (des pages JSP), et de la logique métier (le modèle).

Struts fait partie du groupe ouvert « Jakarta Project » dont le but est de fournir des outils orientés Application Web, basés sur la technologie Java .

### Architecture Struts MVC2



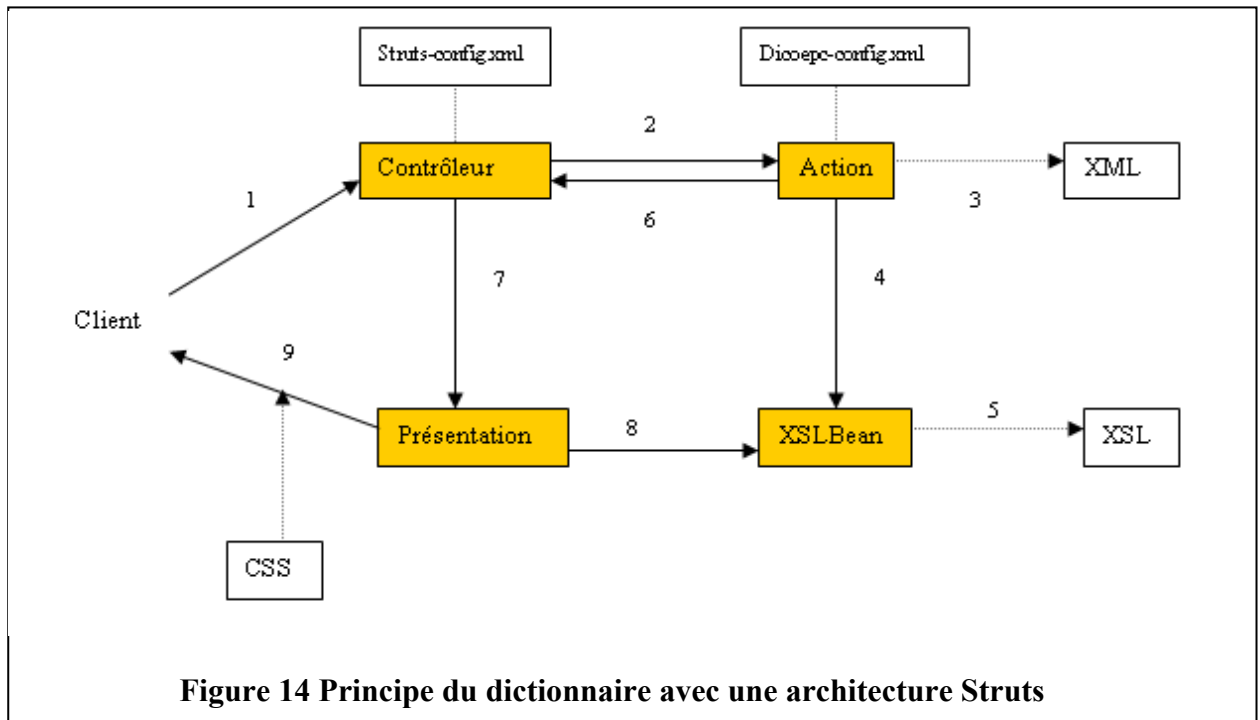
Fonctionnement global de Struts :

1. Le client effectue une requête (via un formulaire par exemple)
2. Le contrôleur, configuré par le fichier struts-config.xml affecte le FormAction avec les données saisies dans le formulaire
3. Le contrôle donne la main à Action
4. Action effectue des traitements sur le FormAction ou sur autre source de données
5. Action rend la main au contrôleur en lui spécifiant la réussite ou non de ses traitements et la page de présentation à appeler
6. Le contrôleur appelle la page de présentation JSP
7. La page JSP consulte les données du FormAction et les formate
8. La page JSP envoie une réponse au client.

## Conception du Dictionnaire

### Principe du Dictionnaire EPC sous STRUTS

#### Les différentes étapes lors d'une requête utilisateur



1. L'utilisateur exécute une requête.
2. Le contrôleur s'occupe d'orienter la requête vers le BeanAction adéquat
3. Le BeanAction récupère les fichiers XML nécessaires
4. Le BeanAction envoie ce résultat au XSLBean pour transformation
5. Le XSLBean récupère le fichier XSL
6. Une fois que le traitement est terminé, le BeanAction rend la main au Contrôleur
7. Le contrôleur rend la main à la page de présentation JSP adéquate
8. La page JSP récupère le résultat du XSLBean
9. La page JSP envoie le résultat à l'utilisateur

#### Fonctionnement des composants

### Le contrôleur

Le contrôleur est un composant STRUTS qui est configuré par le biais du fichier struts-config.xml.

Étapes :

1. il reçoit les requêtes utilisateur
2. il donne la main au BeanAction associé à la requête
3. une fois le BeanAction exécuté, il donne la main au fichier JSP de présentation associé

### Le bean Action

Le bean Action est une classe Java qui s'occupe d'effectuer les traitements associés au modèle.

Dans le Dictionnaire EPC, le bean Action :

1. charge les fichiers XML nécessaires
2. calcule un objet DOM contenant toutes les informations nécessaires à l'affichage :
  - a. Données de traduction
  - b. Informations nécessaires à l'affichage en liste
  - c. Informations nécessaires à l'affichage détaillé
3. envoie cet objet au XSLBean pour transformation
4. met le résultat de la transformation en session, conformément à la norme Struts, pour être accessible par les pages JSP.
5. signale au contrôleur si l'opération s'est bien déroulée ou non

### La Présentation

La présentation est un ensemble de fichiers JSP destinés à la mise en forme de la Vue. Dans la norme STRUTS, ces fichiers effectuent tous les traitements de mise en page. Pour le dictionnaire EPC, c'est la partie XSLBean qui s'en occupe. Ici les fichiers de présentation JSP ont pour seule fonction de récupérer le résultat calculé par le XSLBean et de l'afficher.

## Le XSLBean

Le XSLBean est un bean de transformation XSL. Il reçoit en entrée un objet DOM et une source XSL. Il s'occupe de transformer l'ensemble en un résultat HTML. Il place le résultat en session, conformément à la norme STRUTS pour qu'il soit accessible par les fichiers JSP.

## **Communication entre les composants**

---

L'une des particularités de Struts est de créer une seule instance de chaque classe Java. Cette instance est partagée entre les différents utilisateurs. Il est possible d'utiliser des variables d'instance, mais ces dernières sont partagées entre les différents utilisateurs.

De plus, le seul moyen de communication entre les pages JSP de présentation et les BeanAction est de passer des données dans la session.

### **Mise en place de paramètres dans la session utilisateur**

Les paramètres passés en session sont :

- √ Les résultats envoyés par le XSLBean aux pages JSP
- √ Les objets correspondant à chaque fichier de donnée. Ainsi, chaque fichier n'est chargé qu'une seule fois pour une session donnée, ce qui optimise certains calculs (par exemple, le fichier Traitements.SMArt qui est utilisé par les sections 'données entrées', 'capteurs', 'données élaborées', 'sorties', et 'modèles' ne sera chargé qu'une seule fois pour tous les calculs)
- √ Les objets contenant l'ensemble des données nécessaires à l'affichage de chaque section (appelés ROOT). Ceci permet d'effectuer le calcul une seule fois pour un onglet donné.
- √ Un ensemble de variables de session (langue, ligne, serveur, etc.)

### **Communication inter Actions**

L'absence de variables d'instances implique que l'appel de chaque fonction se fasse en passant un certain nombre de variables en paramètre.

Toutes les fonctions auront au moins comme paramètre :

- √ HttpServletRequest. Cette donnée permettra de retrouver la session utilisateur
- √ String CLE. Cette donnée permet d'identifier la section en cours, et de rechercher les données voulues dans la session, ainsi que les fichiers XSL et SMArt voulus.

## Configuration de l'application Dictionnaire

---

### Dicoepc-config.xml

Le fichier dicoepc-config.xml offre la possibilité de modifier certains paramètres de l'application Dictionnaire sans modifier la code. Ce fichier sert à alimenter la classe Constants.java qui contient toutes les constantes utilisées dans l'application. Ce fichier est stocké dans le répertoire de travail de l'application (WorkingDir).

Il permet de paramétrer :

- √ le répertoire contenant les fichiers XSL
- √ le répertoire contenant la liste des modèles et les données liées aux modèles
- √ divers paramètres (extension des fichiers XSL, des fichiers XML et des fichiers SMArt)

```
<?xml version='1.0' encoding='UTF-8'?>
<Params>
  <Param id="RepModeles">http://web-di/epc/dico/modeles/</Param>      <!-- URL contenant les fichiers modèles -->
  <Param id="RepXML">http://web-dicoepc-dev/xml/</Param>             <!-- URL contenant les fichiers de paramètres
(seveursEPC.xml et traduction.xml)-->
  <Param id="RepXSL">http://web-dicoepc-dev/xsl/</Param>             <!-- URL contenant les fichiers XSL -->
  <Param id="ExtensionSMArt">.<SMArt></Param>                         <!-- extension des fichiers de données (SMArt)-->
  <Param id="ExtensionXML">.<xml></Param>                             <!-- extension des fichiers de
paramètres (seveursEPC.xml et traduction.xml) -->
  <Param id="ExtensionXSL">.<xsl></Param>                             <!-- extension des fichiers de présentation xsl -->
</Params>
```

## ServeursEPC.xml

Le fichier serveursEPC.xml contient toutes les données liées aux serveurs et lignes pour lesquels l'application Dictionnaire peut consulter et afficher les configuration

Il est stocké dans le répertoire de travail de l'application dictionnaire (WorkingDir).

```
<Serveurs>
  <Serveur id="Nom contenu dans la combo">
    <Adresselp>tes-epc01</Adresselp>
    <Port>666</Port>
    <Lignes>
      <Ligne>MtGA3</Ligne>
      <Ligne>Solmed</Ligne>
    </Lignes>
  </Serveur>
```

## Traduction.xml

Le fichier traduction.xml contient toutes les données multilingues, c'est à dire les titres de section, les menus, les titres de tableaux, etc. Chaque donnée est connue en français, anglais, espagnol, et allemand.

Il est stocké dans le répertoire de travail de l'application dictionnaire (WorkingDir).

## Application de Contrôle

L'application de contrôle permet de contrôler des critères essentiels au bon fonctionnement d'une application EPC M2 : existence des fichiers de configuration, validité des fichiers XML, intégrité et cohérence des données saisies.

### Principe

La classe de base **ControleDictionnaire** s'occupe de :

- √ récupérer les paramètres passés dans la requête
- √ charger tous les fichiers à analyser
- √ appeler les différentes classes qui effectuent l'ensemble des contrôles
- √ transformer les informations résultats des tests (contenues dans un objet DOM) en code HTML.

### Les différentes classes

Les classes de contrôle utilisées sont :

- √ **ControleCapteur**. Cette classe teste la cohérence du fichier "cpateurs" avec le fichier "InfosDonnees", le fichier "DonneesEntrees", et le fichier "Traitements"
- √ **ControleDonneeEntree**. Cette classe teste la cohérence du fichier "DonneesEntrees" avec le fichier "InfosDonnees" et le fichier "InfosDonnees"
- √ **ControleInfoDonnee**. Cette classe teste la cohérence du fichier "InfosDonnees" avec le fichier "Traitements"
- √ **ControleModele**. Cette classe teste et la cohérence entre les fichiers "modeles" et "InventaireModeles. Pour chaque modèle déclaré, on testera la cohérence avec le fichier "paras".
- √ **ControleServeur**. Cette classe contrôle l'existence du serveur, de la ligne, et des fichiers. Elle vérifie ensuite si les fichiers respectent bien la norme XML, puis si les fichiers capteurs, infosdonnees, donneesentrees, descriptionligne et traitements ont une structure respectant le schéma XSD associé.

- √ **ContrôleTraitement.** Cette classe teste la cohérence du fichier "Traitements" avec le fichier "paras" et le fichier "InventaireModeles"

## Fichier résultat

Les résultats calculés sont stockés temporairement dans un objet DOM dont voilà la structure :

```
<Result>
  <Section id="donneesentrees">                                //La section à tester (=Onglet)
    <Titre>Données d'Entrée</Titre>                            //Le titre de la section
    <Erreurs>                                                  //Section de résultats des tests
      <Erreur res='true'>                                       //Résultat d'un test (avec le résultat du test dans attribut res)
        <Texte></Texte>                                         //Libellé du test effectué
        <Tooltip></Tooltip>                                     //Explication apparaissant au survol
      </Erreur >
    </Erreurs >
  </Section>
</Result>
```

Une fois tous les calculs effectués, l'objet DOM contenant les données de résultat est transformé en code HTML par le XSLBean qui s'occupe d'appliquer un fichier XSL à l'objet.

## Les apports du stage

### **Pour USINOR SI**

---

Le projet a été terminé dans les délais. Toutes les contraintes de départ ont été respectées, si bien que le résultat du travail effectué pendant le stage a été mis en production chez les clients.

L'application Dictionnaire satisfait pleinement les attentes des clients, si bien qu'il suscite déjà des nouveaux besoins :

- √ Description en langage naturel des traitements d'élaborations
- √ Modification des paramètres de l'EPC M2 à partir de l'interface du Dictionnaire

### **Pour le Stagiaire**

---

Le stage a été très enrichissant, car j'ai eu la chance d'évoluer dans une équipe informatique qui tient à utiliser au maximum les technologies récentes.

#### **Communication**

Dans une grande structure comme Usinor SI, la communication entre les personnes d'un même projet est très importante, surtout quand l'ensemble de ces personnes se trouvent éloignées géographiquement (une partie de l'équipe est à Paris la Défense, et le reste à Sollac Fos). Ainsi, tous les travaux que j'ai effectués ont été documentés, ceci aussi dans le but de faciliter la maintenance. J'ai aussi dû faire des retours de connaissances pour les nouvelles technologies comme Struts que j'ai expérimentées.

#### **Techniques**

Au point de vue technique, ce stage m'a permis d'approfondir une grande partie des notions vues en cours. J'ai ainsi pu découvrir une nouvelle application de Java avec XML et XSL.

Un des aspects intéressants du stage est que l'on m'a orienté vers l'utilisation du Framework Struts appliqué à la méthode de conception web MVC2. Ceci pour permettre à Usinor SI d'avoir un aperçu de ses possibilités.

Il m'a donc fallu non seulement me former aux technologies XML, XSL et leur manipulation avec Java, mais aussi « partir à la découverte » du monde Struts et de ses composants annexes, qui, au début de mon stage étaient encore en version de test.

### **Culture**

Du point de vue culturel, mon stage a été très enrichissant. Le projet Dictionnaire s'inscrit dans un milieu industriel de pointe : la sidérurgie. J'ai été en contact très proche avec tout le processus sidérurgique : de la réception des matières premières à la galvanisation des produits finaux.

Intégrer une grande industrie d'envergure mondiale m'a permis de voir fonctionner un grand groupe de l'intérieur, avec ses avantages et ses inconvénients, et de découvrir la « culture d'entreprise ».

### **Conclusion**

Un stage ne se résume pas à la seule réalisation d'un projet. Il s'agit d'exercer une activité au sein d'une entreprise, c'est à dire qu'il faut s'intégrer à une équipe de travail et respecter l'esprit de production qui l'anime. Tout au long du stage, j'ai bénéficié de très bonnes conditions de travail : la sympathie et la disponibilité des personnes qui m'ont encadré. Cet apprentissage ne peut pas faire l'objet d'un cours, et c'est pour cela que les stages en entreprise sont indispensables dans une formation universitaire.

## Conclusion

Il s'agit de ma troisième expérience en matière de stage, et c'est celle qui aura été la plus instructive, autant sur le plan technique que sur les plans culturels et personnels. Durant cette période d'apprentissage du monde du travail, ma soif d'apprendre et de réussir aura pallié à mon manque d'expérience.

J'ai eu la chance d'évoluer dans une équipe où le stagiaire n'était pas vu comme une contrainte ou comme un employé inexpérimenté, mais comme une personne dynamique pouvant apporter quelque chose. C'est pour ces raisons que l'on m'a confié la réalisation du Dictionnaire et l'exploration des possibilités offertes par des nouvelles technologies comme Struts.

L'autonomie que j'ai eue durant mon stage m'a permis de commettre des erreurs et d'en tirer des leçons.

Ce stage a réveillé mon envie d'entrer dans la vie active, et ma formation scolaire m'a donné tous les atouts pour réussir.