



# ANALYSE ET DEVELOPPEMENT D'UNE APPLICATION DE GESTION DE FOURS A COKE : APPLICATION A CARSID

Caroline Herssens, Manuel Kolp, Yves Wautelet

IAG –Institut d'Administration et de Gestion, ISYS- Unité de systèmes d'information, Université Catholique de Louvain,  
1 Place des Doyens, 1348 Louvain-la-Neuve, Belgique

Email: {herssens, kolp, wautelet}@isys.ucl.ac.be

**Abstract:** Le monde économique actuel consacre à « l'information » une valeur déterminante. En effet, même pour des entreprises du secteur secondaire (activités productrices de matières transformées) comme une entreprise sidérurgique, les flux d'informations concernant la concurrence, les indicateurs économiques, l'état du marché ou plus simplement la production doivent être exploités, traités, formatés et compris pour une prise de décision optimale. Dans cette optique, l'entreprise sidérurgique Carsid travaille depuis quatre ans en partenariat avec le département de recherche ISYS de l'IAG afin de moderniser ses systèmes d'information. Ce document présente l'analyse et le développement d'une application de gestion des fours à coke dont l'objectif principal est le *reporting* des pannes pouvant survenir sur ceux-ci. Au départ de données relevées en temps réel sur les fours et autres machines lors des enfournements, cuissons et défournements, l'application peut diagnostiquer certains dysfonctionnements, en faire le rapport auprès de différents corps de métier et proposer des solutions concrètes pour y remédier. Nul doute qu'un tel outil peut s'avérer de première importance pour assurer la pérennité de l'appareil de production et d'exploiter celui-ci de manière optimale. Enfin, notons également que cette application « fours » n'est qu'un module d'un projet de modernisation et d'exploitation des systèmes d'information beaucoup plus global qui instancie le processus général décrit par le *Unified Process*.

## 1. INTRODUCTION

La situation économique actuelle oblige les entreprises, quel que soit leur domaine d'activité, à demeurer compétitives sur des marchés de plus en plus concurrentiels. Pour ce faire, celles-ci doivent mettre tout en œuvre pour réduire leurs coûts et ce notamment en passant par l'utilisation des nouvelles technologies de l'information et de la communication. Celles-ci, bien que représentant souvent un coût d'installation important, permettent d'augmenter rentabilité et productivité. Le monde sidérurgique n'échappe pas à la règle, bien au contraire, secteur porteur il y a quelques décennies il est maintenant sujet à une concurrence accrue des pays à main d'œuvre peu coûteuse et, en Belgique, les fermetures de sites se multiplient. C'est pourquoi il est important pour une entreprise de ce secteur de profiter de ce qu'offrent les nouveaux développements technologiques pour tenter d'assurer sa pérennité sur le marché. C'est dans cette volonté qu'est née la collaboration entre l'unité des systèmes d'information (ISYS) de l'institut d'administration et de gestion (IAG) de l'université catholique de Louvain (UCL) et

l'entreprise sidérurgique Carsid. L'application de travaux de recherche et des méthodologies d'ingénierie informatique les plus récentes permet à cette entreprise d'améliorer sa productivité et en conséquence sa rentabilité. C'est dans ce contexte que s'inscrit ce travail, il est la quatrième étape d'un projet entrepris voici quatre ans. La première consistait en une première modélisation des processus de production de la cokerie (voir [DFK+03]). Le second effort visait le rétro ingénierie des bases de données relationnelles existantes en vue d'une exploitation optimale des sources d'information (voir [FIK+03]). La troisième itération s'inscrivait dans la continuité du travail entrepris et a permis de compléter la première modélisation à la lumière des nouveaux besoins et des remarques émises par le personnel de la cokerie (voir [WLK04]). D'autre part, les processus de gestion des entrées et des sorties de marchandises par camion (nommé bascule) et par rail (nommé traction) ainsi que la gestion des stocks qui en découle ont également été modélisés (voir [WLK04b]). Cette quatrième étape consiste une nouvelle fois à compléter les travaux antérieurs. En effet, lors du développement de modules logiciels liés aux besoins identifiés dans les phases précédentes, le besoin d'une application permettant d'établir un diagnostic sur les fours de la cokerie lorsque ceux-ci présentent des

problèmes au défournement s'est fait ressentir. Lorsqu'un four connaît un problème important et qu'il doit être réparé, la mise en veille de la production de coke qui en découle induit des coûts colossaux pour l'entreprise. En effet, la perte de production liée à son inactivité a pour conséquence une fourniture moindre en coke au haut fourneau qui ne pourra en conséquence plus fonctionner de manière optimale. Si l'on y adjoint le coût de la réparation proprement dite et le coût lié à son rallumage il semble évident qu'il est essentiel d'anticiper au maximum les pannes pouvant survenir. C'est pourquoi les moindres anomalies observées au défournement doivent être analysées et interprétées afin de permettre une intervention brève du service d'entretien selon un planning déterminé plutôt que de devoir effectuer des réparations dans l'urgence. L'application développée lors de cette quatrième itération comporte un ensemble de fonctionnalités qui permettront à l'utilisateur de gérer différents aspects du contrôle des fours et notamment leurs entretiens.

Une analyse concernant la mise en place d'une telle application dans le contexte de la cokerie doit préalablement être réalisée. Celle-ci sera effectuée à l'aide de divers outils de modélisation : le langage UML, standard de l'industrie et déjà utilisé dans les itérations précédentes du projet, mais également la méthodologie KAOS qui vient au mieux compléter un langage UML parfois imparfait et limité (voir [RJB98]). Les modèles développés, mènent à l'implémentation d'une application de gestion des fours qui interagit avec les bases de données existantes de l'entreprise (reposant sur le système de gestion de base de données *SQL Server 2000*). Le langage de programmation *Visual Studio.NET*, standard utilisé par le client CARSID pour le développement de ses applications de gestion a été adopté. Le logiciel va permettre, via la centralisation des données concernant les défournements, d'analyser, en vue de résoudre, les divers problèmes apparaissant sur les fours. Il pourra être utilisé à la fois par l'ingénieur de production pour diagnostiquer les divers dysfonctionnements, planifier les entretiens ou comparer divers défournements mais aussi par le contremaître chargé du bon déroulement des défournements qui veillera à notifier les anomalies.

## **2. L'APPLICATION FOUR : UN MODULE LOGICIEL DÉVELOPPÉ DANS LE CADRE D'UN PROJET UP**

Comme précisé précédemment, le présent travail s'inscrit dans un projet visant à améliorer la gestion de la production, le traitement de l'information, l'automatisation des processus, etc. de la cokerie de Carsid. Plusieurs étapes préalables à cet effort ont déjà été réalisées, celles-ci peuvent être replacées dans le contexte du Unified Process ou UP (voir [JB00, JBR99, KRU03, RUP03]), méthodologie de gestion de projet qui permet d'encadrer la conception et le développement de logiciels faisant intervenir l'utilisateur de façon intensive. Le UP est composé de disciplines exécutées séquentiellement (axe vertical) et répétées de façon itérative au sein de différentes phases.

Les disciplines telles que définies par le UP sont de deux types : six principales (modélisation métier, gestion des exigences, analyse et conception, implémentation, test et déploiement) ainsi que de trois dites « de support » (gestion de la configuration et des changements, gestion de projet et environnement). Celles-ci sont exécutées séquentiellement (axe vertical) et répétées itérativement (axe horizontal) ; chaque itération étant comprise dans une phase. Les phases, aux nombres de quatre, se distinguent par les « points d'ancrage » donnés dans le cycle de vie du projet (situés à la fin de certaines itérations) où l'on va faire le bilan du travail accompli et du travail restant à accomplir (voir [Boe96]). Ainsi, tout projet commence par une phase d'*Inception*, se poursuit par la phase d'*Elaboration*, la phase de *Construction* et enfin la phase de *Transition*. Chacune de ces phases est composée d'une ou de plusieurs itérations. Notons également que toutes les disciplines ne sont pas abordées avec la même intensité lors des différentes itérations du projet comme exprimé dans le profil générique du RUP en figure 1.

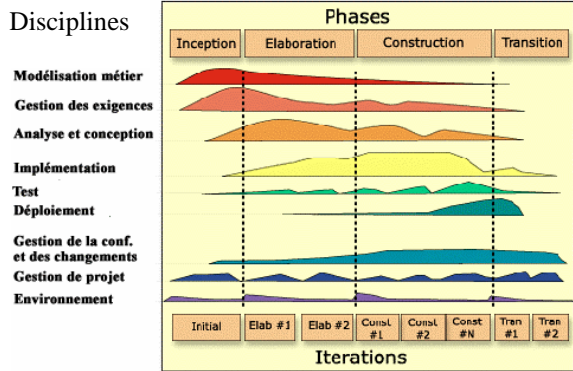
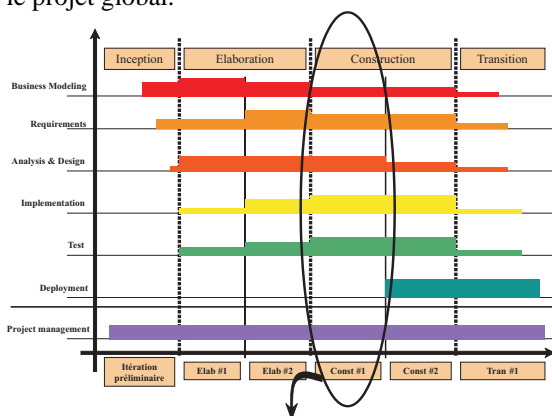


Figure 1 : Itérations du UP

Le projet d'informatisation de la cokerie de Carsid instancie le UP (pour plus d'informations à ce sujet, voir [WLK04, Wau03]), le présent travail est l'une de ses itérations (voir Figure 2). Il est dès lors intéressant de replacer la présente itération dans un contexte plus global. Celle-ci constitue la première itération de la phase de construction. En effet, différentes itérations ont déjà permis de dégager la plupart des besoins des stakeholders (utilisateurs finaux, ingénieurs, management, ...) et le travail se porte essentiellement sur un raffinement de l'implémentation de besoins précédemment identifiés. L'outil de reporting sur les fours se fonde sur le besoin d'une application de gestion efficiente des fours et raffine celui-ci principalement au niveau du système de diagnostic des pannes. Cette itération cadre donc dans un effort plus global, en conséquence la modélisation effectuée s'intègre et complète les modélisations précédentes. L'entièreté des modèles développés dans le cadre de cette itération ainsi que leurs spécifications complètes peut être trouvée dans [DH05]. Le lecteur doit néanmoins garder à l'esprit leur intégration dans le projet global.



Itération développée dans ce travail

Figure 2 : Instanciation du schéma général du UP au présent projet

### 3. CONTEXTE DE TRAVAIL : LA PROBLÉMATIQUE FOURS

Cette section se consacre à l'analyse des problèmes pouvant survenir sur les fours. Le lecteur désireux de replacer cette problématique dans le processus de production de la cokerie se référera à [WLK04].

Lors d'un défournement, de nombreuses forces sont mises en cause par la défourneuse, si celles-ci ne sont pas gérées correctement, les fours peuvent être endommagés. Des réparations impliquent des arrêts de production en plus du coût induit par la réparation elle-même. Les causes de cette dégradation des cellules de fours peuvent être multiples, il est donc indispensable de les identifier pour tenter de les éviter.

Les principales causes de problèmes sur les fours sont :

- apport de chaleur excessif ou insuffisant
- dégradation des parois réfractaires des fours
- défaut de cokéfaction
- défaillance mécanique ou électrique sur les machines

Celles-ci mènent à la nécessité d'exercer des énergies plus fortes sur les machines de défournement ou même à des blocages de fours (le coke ne parvient pas à sortir de la cellule du four) qui peuvent conduire à un dégagement manuel de ceux-ci, entraînant des pertes de temps plus ou moins importantes selon les cas.

Etant donné la fréquence des défournements, il est presque impossible de vérifier l'état de chaque four avant son remplissage, en effet, ces vérifications comprennent les éléments suivants :

- inspection du côté machine
- inspection du côté coke avec déplacement obligatoire du guide coke
- inspection de la sole du four
- inspection de la voûte depuis le dessus des fours puis de la défourneuse

L'inspection des fours évite des problèmes graves qui conduisent à une dégradation puis à un remplacement des fours. Un suivi des courbes d'intensité et des autres données fournies permet la détection et la prévention de ces différents problèmes.

Outre l'état des fours et des machines, d'autres facteurs entrent en compte. C'est le cas notamment de la composition du coke qui dépend de l'origine du mélange des différents charbons utilisés ou encore les conditions de cuisson. Pendant la cokéfaction, le coke se rétracte et crée un décollement par rapport aux parois du four. En fin de cuisson, le pain de coke doit logiquement être décollé des parois du four et présenter une cohésion suffisante pour supporter son propre

pois. Pour défourner ce saumon de coke, une certaine force doit être appliquée à celui-ci par le bélier de la défourneuse pour vaincre les forces de frottement sur la sole du four.

### 3.1 Problèmes des fours détectés à l'aide de la puissance au défournement

Les données de puissance au défournement ne sont actuellement disponibles que pour les batteries Koppers et Didier étant donné que seule la défourneuse de ces batteries de fours utilise une mesure d'intensité. Celle-ci a un moteur fonctionnant avec du courant alternatif ; la vitesse en charge du moteur de la crémaillère est constante, contrairement à celle de la défourneuse servant sur Coppée 1 et 2. Une limitation de la puissance est donc nécessaire. En effet, s'il n'y a pas de décollement durant la compression, une partie de la pression exercée par le bélier se transfère par réaction via le saumon de coke sur les parois du four. Si la puissance n'est pas limitée, de graves déformations peuvent alors apparaître sur le four. Il existe donc un seuil limite qu'il ne faut pas dépasser. Sur les autres batteries, un système similaire a été mis en place permettant de tirer des conclusions assez semblables à celles valables pour la puissance au défournement.

L'analyse des différents problèmes pouvant survenir lors d'un défournement s'appuie avant tout sur un ensemble de valeurs moyennes récoltées qui permettront de détecter des anomalies. Le suivi des courbes d'intensité du moteur de la défourneuse a permis de diagnostiquer différentes causes d'éventuelles difficultés de défournement ainsi que d'analyser les tendances générales en fonction des comportements de marche (voir [CB96]).

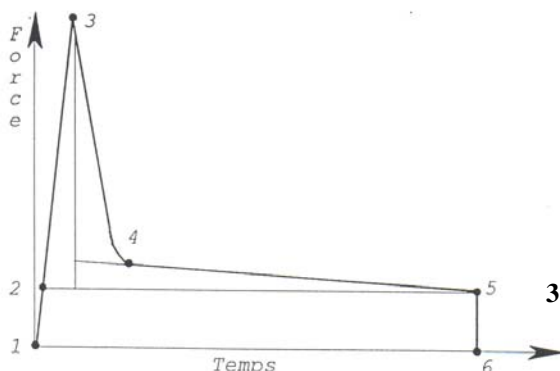


Figure 3: Force appliquée par la défourneuse lors d'un défournement

La force appliquée lors d'un défournement par le moteur de la défourneuse est représentée sur le

schéma de la figure 3. La première partie de la courbe représente la force exercée lors de l'entrée du bélier de la défourneuse dans le four. La seconde partie de la courbe représente l'énergie dépensée pour vaincre les forces de frottement du saumon de coke sur les parois du four. Ensuite, une fois le saumon de coke décollé des parois du four, la force exercée décroît rapidement car la conicité des parois du four entraîne une disparition des forces de frottement. Pour la suite, la force exercée reste proportionnelle à la quantité de coke restant à pousser hors du four et ce jusqu'au retour de la crémaillère.

#### 3.1.1 Paramètres influençant la puissance de défournement

De nombreux paramètres d'exploitation de la cokerie ont une influence sur la force à mettre en œuvre lors du défournement. Ces catégories se classent en quatre classes bien distinctes :

##### 3.1.1.1 Causes liées au défournement du coke

Celles-ci dépendent avant tout de la nature de la pâte à coke, de l'enfournement et des conditions de cuisson :

- retrait latéral du saumon insuffisant en fin de cuisson : le coke est en contact avec les parois du four avant compression suite à une irrégularité lors de l'enfournement qui a provoqué un retrait variable d'un endroit à l'autre du saumon ;
- fragilité du pain de coke : l'empilage de coke est disloqué par la compression, ce qui entraîne d'importantes forces de frottement sur les parois ;
- structure non uniforme de coke : certaines zones du coke sont fragilisées par la suite de défauts de chargement ou d'hétérogénéité de chauffage ;
- niveau de coke trop élevé dans le four : la partie supérieure de la cellule est obstruée par un mauvais chargement ;
- collage du coke sur la sole du four : suite à un ramollissement des cendres à cause d'une surchauffe du voisinage de la sole ou à un dégraphitage de la sole avant enfournement.

##### 3.1.1.2 Causes liées aux fours

Les différents problèmes de défournement dont la responsabilité incombe aux fours sont :

- état de la sole : avec des fours endommagés, le frottement le long des parois est plus élevé et a donc une influence immédiate sur la puissance nécessaire lors du défournement ;

- dégraphitage de la cellule : qui apparaît suite à une réparation de la cellule ou suite à un reclassement dans les séquences de défournement ;
- étanchéité de la cellule : toute entrée d'air parasite se traduit par une anomalie au niveau des gaz et par la formation de quantités importantes de graphite sur les parois du four ;
- déformation des cadres de porte ou des entrées de four.

### 3.1.1.3 Causes liées aux conditions de chauffage

Les problèmes pouvant être rattachés à de mauvaises conditions de chauffage sont :

- température moyenne de la batterie ;
- hétérogénéités locales de température ;
- hauteur de flamme ;
- durée de cuisson ;
- changements de gaz combustible.

### 3.1.1.4 Causes liées aux machines

Les problèmes de défournement inhérents aux différentes installations mécaniques utilisées lors d'un défournement sont les suivants :

- défourneuse : des irrégularités peuvent survenir dans les courbes d'intensité et se traduire par des consommations énergétiques élevées ou par un coincement de la crémaillère ;
- guide-coke : peut être obstrué par la suite de défauts d'alignement de l'auge mobile.

Les causes d'anomalies liées aux machines sont en principe facilement détectables.

## 3.1.2 Quelques courbes de puissance de défournement expérimentales

### 3.1.2.1 Courbe d'intensité à vide

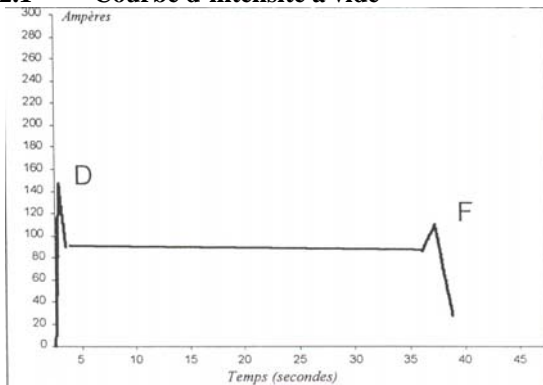


Figure 4: Force appliquée sur un four vide

Lorsque la crémaillère est utilisée dans une cellule vide, les variations d'intensité suivantes sont observées : le premier pic correspondant à

l'intensité lors du démarrage du moteur tandis que le second correspond au freinage précédant le retour de la crémaillère.

### 3.1.2.2 Courbe normale

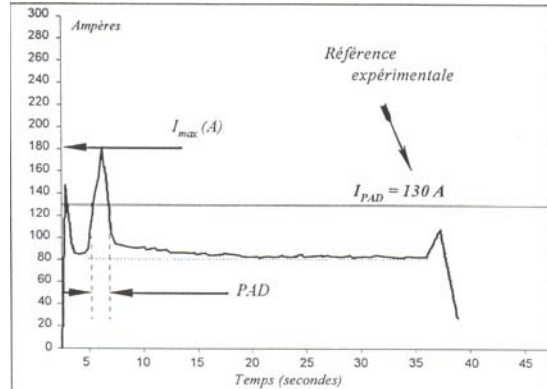


Figure 5 : Force appliquée sur un four sans anomalies

Le profil d'une courbe lors d'un défournement où aucun incident ne survient montre les différentes phases pouvant être observées :

- la phase de compression ;
- la phase de décollement (à laquelle correspond l'intensité maximale =  $I_{max}$ ) ;
- la diminution du frottement global ;
- la diminution de l'intensité linéaire.

Deux paramètres sont à prendre en compte pour l'analyse des problèmes : la largeur du pic au défournement (PAD) et l'intensité maximale déployée par la défourneuse lors du défournement du four ( $I_{max}$ ). A chaque anomalie correspondent des valeurs particulières de ces deux paramètres. Par exemple ; un défournement normal donnera une  $I_{max}$  de l'ordre de 160 ampères maximum et un PAD de l'ordre de 1 à 2 secondes selon la cellule. La durée totale du défournement est quant à elle de l'ordre de 37 à 38 secondes.

### 3.1.2.3 Blocage de four

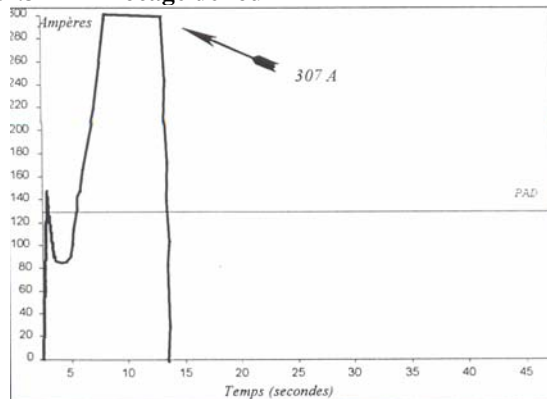


Figure 6 : Force appliquée sur un four bloqué

Lorsque le saumon de coke ne se décolle pas correctement des parois du four, l'intensité

maximale mise en œuvre est dès lors supérieure à 160 ampères, la défourneuse s'arrête et le four se bloque. Sur la figure, le blocage du four est caractérisé par une  $I_{\max} = 307$  ampères suite à la limite de l'échelle utilisée.

### 3.1.2.4 Irrégularité de chauffage

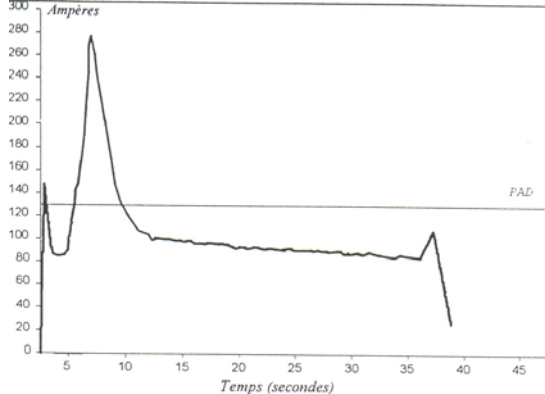


Figure 7 : Force appliquée sur un four avec irrégularité de chauffage

Les défournements sur lesquels apparaissent des anomalies de chauffage sont caractérisés par une  $I_{\max}$  supérieure à 200 ampères et un PAD de l'ordre de 2 à 4 secondes. Le problème se présente soit sur l'entière du four (changement de gaz) soit sur une zone particulière du four, au niveau d'un des deux piédroits. Si le problème se situe sur une partie du four uniquement, il faut également s'attendre à une anomalie sur la courbe du four voisin. Si il n'y a pas d'anomalie sur un des fours voisins, il peut s'agir alors d'un problème au niveau du temps de cuisson. Une autre cause pour ce type de courbe peut être un dégraphitage important qui représente une difficulté supplémentaire pour décoller le saumon de la sole.

### 3.1.2.5 Défaut de parois

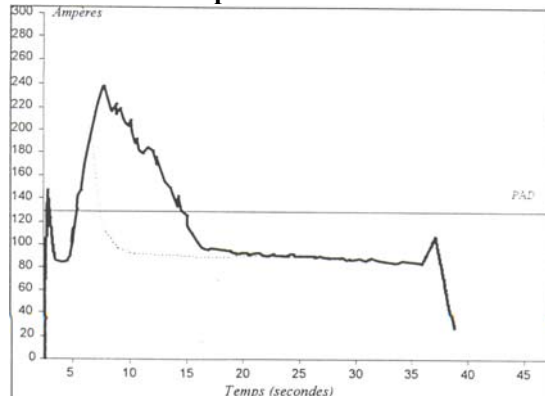


Figure 8 : Force appliquée sur un four présentant des défauts de paroi

Lorsque les parois du côté coke sont dégradées, le PAD peut atteindre 9 à 10 secondes dans le cas d'anomalies graves tandis que l'intensité maximale augmente avec le frottement supplémentaire (accrochages de coke sur la paroi) pour aller jusqu'à 200 ampères.

### 3.1.2.6 Défauts systématiques

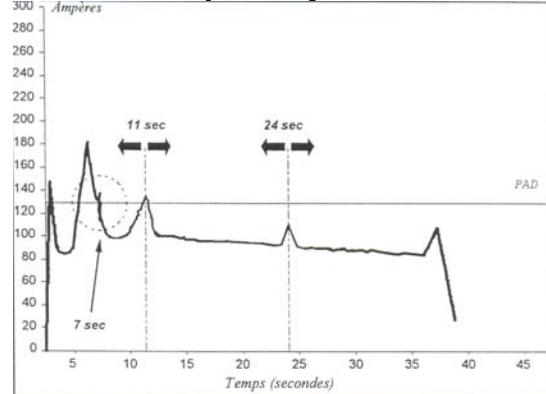


Figure 9 : Force appliquée avec défauts systématiques

Sur certaines courbes, apparaissent des défauts ponctuels et répétitifs. Comme la vitesse est constante pour tous les fours, un défaut au temps  $t$  correspond à un endroit précis de la cellule. Par exemple : l'entrée du poisson dans le four, défaut qui apparaît quand il y a du graphitage à l'entrée de la cellule. Ou encore avec le basculement de la crémaillère sur la sole du four lors du défournement.

### 3.1.2.7 Défaut de cintrage

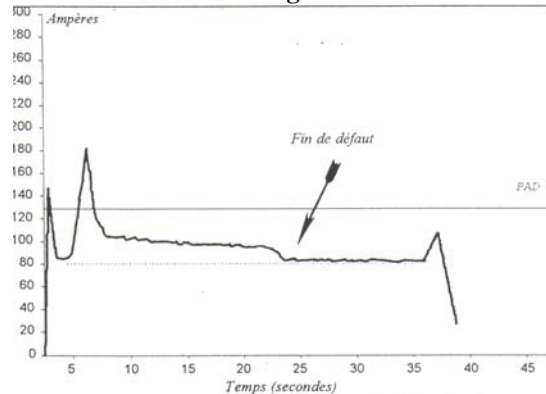


Figure 10 : Force appliquée sur un four présentant un défaut de cintrage

Un défaut de cintrage induit une intensité plus forte que la normale dans la partie linéaire de la courbe de défournement. Cette irrégularité apparaît quand l'encadrement du côté du coke est cintré dans sa partie centrale, dès que le retrait latéral compense ce cintrage, le reste du défournement se produit normalement.



### 3.2 Problèmes des fours détectés à l'aide des irrégularités de chauffage

Les données concernant la puissance au défournement ne sont pas les seules disponibles, les températures de saumon de coke qui sont mesurées lors des défournements sont également exploitables. Cette installation est appelée cotherm et permet de régler la chauffe du charbon car elle reflète exactement la température des carneaux de chauffage.

L'ensemble des batteries de fours de la cokerie sont composées de 126 fours et de 126 piédroits, ces derniers comprenant chacun 26 carneaux qui sont les tubes dans lesquels circulent les gaz afin de chauffer les fours. Les mesures sont réalisées à l'aide de pyromètres situés sur l'auge mobile du guide coke, il y en a 3, situés à différentes hauteurs ; respectivement 0,75m, 1,6m et 3m qui reflètent les températures à diverses hauteurs du saumon de coke.

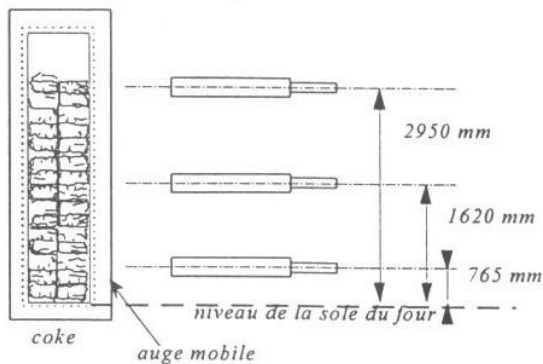


Figure 11 : Pyromètres à 3 niveaux

Lors du défournement, les clapets se trouvant devant les pyromètres s'ouvrent et de l'air comprimé est insufflé sur celui-ci pour éviter que la lentille ne se salisse. Le saumon, poussé par la défourneuse défile devant les pyromètres qui vont enregistrer les températures. Les données de température sont prises pour chacun des 3 pyromètres à 26 reprises pour refléter les températures des 26 carneaux du four.

Ces données sont biaisées, en effet, elles ne reflètent pas exactement la chaleur des carneaux mais celle du saumon de coke à sa sortie ; le coke n'étant pas aussi chaud que les piédroits lors de sa cuisson et étant refroidi lors de son défournement, il faut ajouter aux températures obtenues un delta de température si l'on souhaite obtenir la température réelle des carneaux. Après mesure expérimentale, ce delta de température a été fixé à 150°C.

Actuellement, ces données sont traitées sous la forme de cartes thermiques, celles-ci étant dessinées à partir d'une courbe de Gauss.

Les données les plus proches de la moyenne sont représentées en blanc, celles qui s'en écartent le plus et qui lui sont inférieures se voient attribuer des couleurs se rapprochant de plus en plus du bleu tandis que celles qui s'écartent de manière supérieure se voient attribuer des couleurs se rapprochant progressivement du rouge. Une carte thermique montre alors l'ensemble des fours d'une batterie avec pour chaque carneau la couleur qui lui a été attribuée lors du dernier défournement.

Voici un exemple d'une carte thermique, les colonnes grises étant les fours pour lesquels il n'y a pas de données et les cases mauves, les mesures pour lesquelles des problèmes sont survenus.

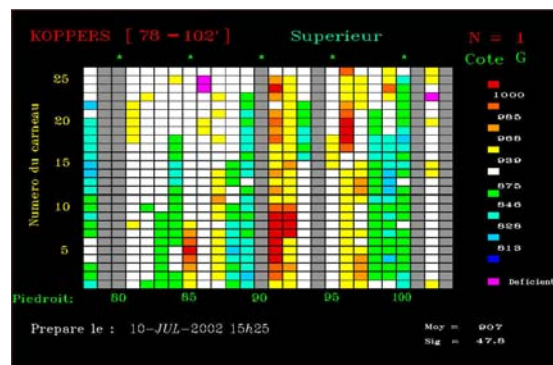


Figure 12 : Carte cotherm

Les différentes données qui sont exploitables sont donc : les différentes températures, les moyennes de températures et les écarts types par rapport aux moyennes.

Ces différentes mesures permettent de déduire si la température globale d'un four est trop basse ou trop haute et également de se concentrer sur une partie de four puisque les températures sont disponibles pour tous les carneaux. Si le problème est local, il sera traité au niveau des carneaux concernés.

Certaines anomalies visibles sur ces cartes permettent également de trouver des disfonctionnements au niveau du matériel :

- mauvais alignement des pyromètres : on observe sur la carte thermique des lignes verticales de différentes couleurs, cela signifie pour les données que les températures relevées pour les différents carneaux d'un four sont les mêmes et qu'aucun écart type n'est observé ;
- pyromètre sale : on observe une température moyenne trop basse car les données relevées par le pyromètre sont altérées.



### 3.3 Problèmes des fours détectés à l'aide des temps de cuisson

Un autre paramètre entre en considération dans la gestion des dysfonctionnements de fours : le temps de cuisson, il est encodé dans la base de données par le machiniste en charge du défournement.

Le processus de cokéfaction du charbon au sein d'un four dure normalement 16h mais pour la batterie Koppers qui est plus vieille, le charbon va cuire à peu près 18h. Ce temps peut aussi varier selon la qualité du charbon, qui avec une granularité plus grosse demande un plus grand temps de cuisson. Il augmente également lorsque la cokerie est en grève, en effet, ceux-ci ne peuvent pas stopper la production mais la diminue en augmentant les heures de cuisson. En règle générale, plus longtemps le charbon va cuire et meilleure sera sa qualité, ce qui pose un dilemme pour l'entreprise qui vise avant tout un objectif de rentabilité et donc de productivité, quitte à cuire un peu moins longtemps mais obtenir de plus grandes quantités de coke.

Si le temps de cuisson n'a pas été respecté, la qualité du coke en sera bien souvent affectée, il est donc impératif que les anomalies lui incombant soient détectées.

### 3.4 Problèmes des fours détectés à l'aide du profil de coke

Le profilocoke est une aide à l'analyse de problèmes des fours ; il fournit le profil du saumon de coke lors de son défournement. Il est mesuré à l'aide de sondes radioactives situées sur le guide coke. Les mesures sont prises grâce à un trou de visée d'un centimètre de large et d'un mètre de long qui se situe sur le haut du four comme représenté à la figure 13.

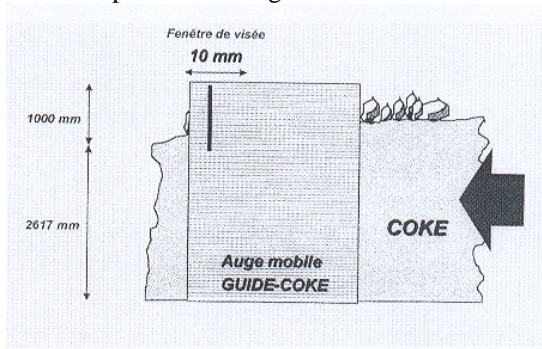


Figure 13 : Fenêtre de visée du profilocoke

Le profil de coke ainsi mesuré va permettre à la fois de détecter les anomalies de cuisson, de calculer le volume de coke obtenu après cuisson et

également de contrôler le chargement des fours. [Fiv04]

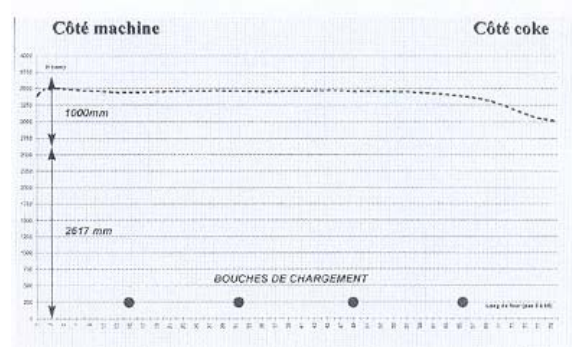


Figure 14 : Fenêtre de visée du profilocoke

En ce qui concerne des anomalies; le profilocoke permet de détecter :

- Un défaut de chargement :

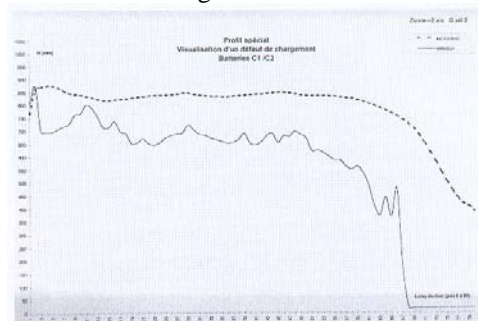


Figure 15 : Défaut de chargement du coke

- Un effondrement :

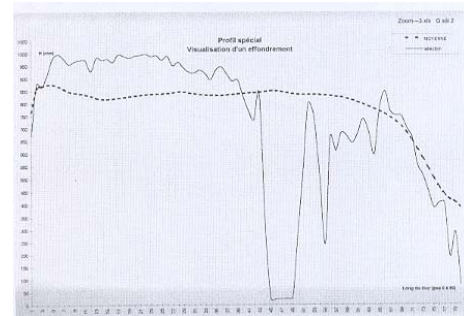


Figure 16 : Effondrement du coke

### 3.5 Processus de diagnostic de dysfonctionnements sur un four

Comme vu précédemment, la cokerie dispose de plusieurs outils permettant de détecter les anomalies lors des défournements, celles-ci permettant de connaître le problème que rencontre le four concerné.

Les principaux problèmes rencontrés sur les fours sont divisés en quatre catégories :

- les problèmes de chaleur
- la dégradation des parois réfractaires

- les défauts de cokéfaction
- les défaillances mécaniques sur les machines

Les outils permettant de détecter ces divers problèmes sont les suivants :

- le machiniste présent lors de chaque défournement qui surveille le bon déroulement de l'opération
- les cartes de températures ou cotherm
- la puissance exercée par la défourneuse lors du défournement
- le temps de cuisson
- le profil du coke lors de son défournement

Chacun de ces outils permet plus spécifiquement de détecter une des catégories de disfonctionnement des fours.

Les problèmes de températures seront mis en évidence par les cartes de températures, la dégradation des parois réfractaires sera détectée à l'aide de la puissance de défournement. Les défauts de cokéfaction seront trouvés à l'aide du profilocoke et de la durée de cuisson, tandis que les problèmes mécaniques seront la responsabilité du machiniste qui veille au bon déroulement des opérations.

### 3.5.1 Considération des différentes données

C'est d'abord l'avis du contremaître qui rentrera en compte, en effet, si un problème technique est détecté sur une des machines servant au défournement il est normal de ne pas tenir compte des différentes données obtenues lors de ce défournement et de réparer la panne observée. Ensuite, c'est la puissance de défournement qui servira au diagnostic ; des problèmes de températures et de profil de coke peuvent apparaître suite aux anomalies qui sont détectées avec la puissance de défournement, il est donc normal de tenir compte de celle-ci en priorité. Par après, c'est le profil du coke qui sera observé ; si celui-ci présente un effondrement, les températures de chauffe moyenne seront affectées, il est donc nécessaire d'observer celui-ci avant. Le dernier outil qui sera examiné est la carte de températures car c'est elle qui est la plus sujette à des modifications suite aux différents problèmes pouvant être rencontrés.

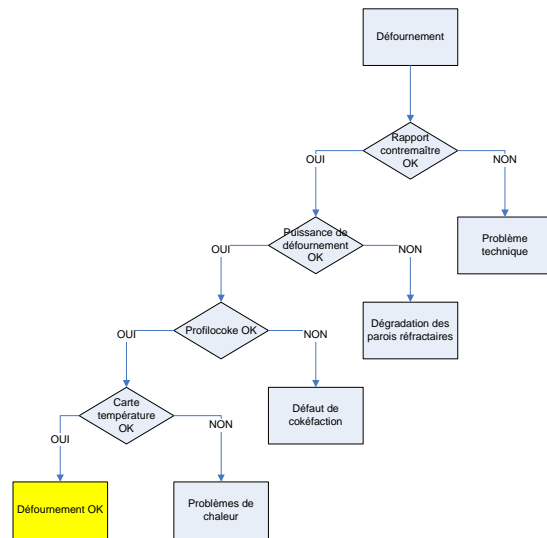


Figure 17 : Processus de décision pour la considération des différentes données

### 3.5.2 Le cokier et la durée de cuisson

La première chose dont il faut tenir compte lors de l'analyse des données de défournement est l'avis rendu par le cokier. Si un problème mécanique s'est déclaré durant la cuisson ou le défournement d'un four, le cokier doit signaler celui-ci. De même, la durée de cuisson est un élément primordial pour la bonne qualité du coke, si celui-ci n'a pas cuit assez longtemps, les autres données seront altérées.

### 3.5.3 Utilisation de la puissance de défournement

Les deux principaux paramètres utilisables de la puissance au défournement sont, comme expliqué avant : l'intensité maximale (imax) et la durée du pic d'intensité (pad). Les différents diagnostics possibles suivent le schéma de décisions de la figure 18 :

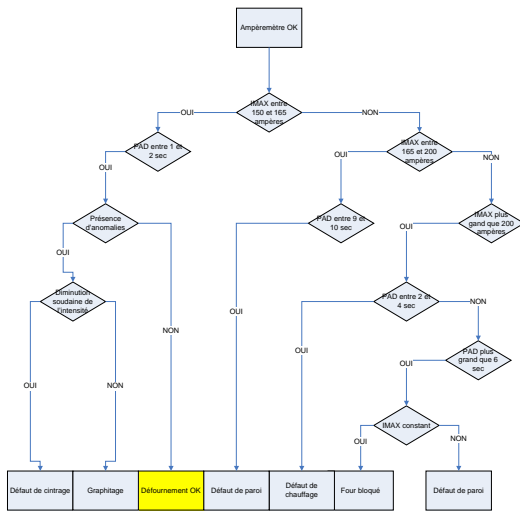


Figure 18 : Processus de décision pour la puissance exercée au défournement

### 3.5.4 Utilisation du profilocoke

Le profil de coke obtenu lors d'un défournement permet de diagnostiquer deux types de problèmes de cokéfaction : un effondrement ou un défaut de chargement. Les schémas de ces deux cas sont présentés plus haut.

### 3.5.5 Utilisation des cartes de températures

Les cartes de température vont permettre de corriger les erreurs de chauffage afin d'améliorer la qualité du coke sur les défournements futurs. Les 'rouges' et les 'bleus' dont on parle dans le schéma sont les couleurs associées aux carneaux, selon le fait que le gaz monte ou descende dans le carneau, on associe à celui-ci une couleur.

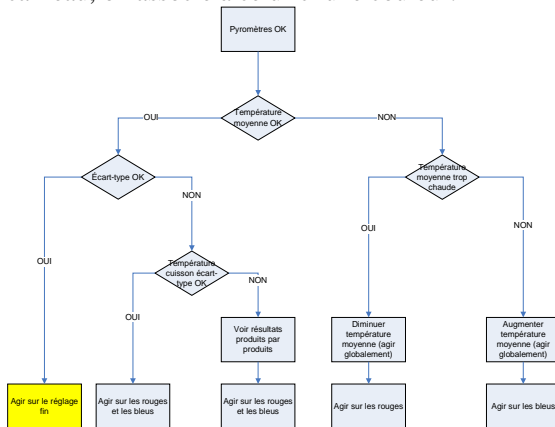


Figure 19 : Processus de décision pour les cartes de températures

## 4. ELEMENTS DE MODELISATION

La modélisation est une des tâches les plus importantes dans le processus de développement d'un système. La phase consacrée à l'analyse peut être considérée comme plus stratégique que celles dévolues à la conception et l'implémentation proprement dites. Il faut en effet fondamentalement représenter, comprendre et identifier les exigences du système afin de concevoir puis d'implémenter une application stable et performante.

Avec l'augmentation de la complexité des systèmes à élaborer, le choix d'une méthode de développement appropriée se révèle primordial pour le succès des travaux. Pour cela, il existe plusieurs orientations disponibles, c'est-à-dire des approches différentes pour comprendre, représenter, analyser et concevoir un système. Le problème sera décomposé en plusieurs modèles différents liés entre eux.

### 4.1 KAOS

La méthodologie KAOS est une méthode d'analyse des besoins de logiciels dirigée par les buts, elle a été mise au point par le professeur A. Van Lamsweerde, UCL. Le modèle des buts tel qu'il est décrit dans KAOS permet de visualiser l'ensemble des exigences attendues du système et donc ce qui devra être réalisé par le logiciel (voir [AvL04]). Pour ce faire, la manière la plus simple de procéder est de partir des buts de haut niveau, qu'ils soient fonctionnels (qui concernent le service) ou non fonctionnels (qui concernent la qualité du service), et de raffiner ceux-ci jusqu'à arriver à des buts pouvant être affectés à des agents (éléments actifs du système) de l'environnement du logiciel ou au logiciel lui-même. De cette manière, les différents agents du système sont identifiés car chacun d'entre eux est responsable d'un seul ou de plusieurs buts terminaux. Pour réaliser le modèle des buts, le logiciel Objectiver (logiciel utilisé pour la méthodologie d'analyse KAOS, disponible à l'adresse [www.objectiver.com](http://www.objectiver.com)) a été utilisé. Ce dernier a été développé par le Cediti (Centre de diffusion des technologies de l'information, société de développements e-business, [www.cediti.be](http://www.cediti.be)).

Ce logiciel utilise pour la notation des buts la syntaxe suivante :

- Les buts sont les objectifs devant être réalisés par le système. Les éléments

principaux du modèle des buts sont représentés de la manière qui suit :



Figure 20 : But

- Les buts qui ne sont pas terminaux, c'est-à-dire qui ne sont pas attribués directement à un agent du logiciel ou de l'environnement vont être raffinés jusqu'à ce que leurs descendants soient attribués à des agents, les raffinements sont représentés comme sur le schéma suivant :

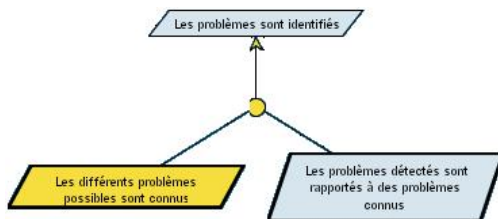


Figure 21 : Raffinement de but

- Les buts terminaux sont des buts dont la réalisation est confiée à des agents, ceux-ci pouvant faire partie de l'environnement ou du logiciel à développer, ils sont représentés de la manière suivante :



Figure 22 : Agent

- Les buts terminaux peuvent être de deux types différents:
  - Les *requirements* qui sont des buts pour lesquels la responsabilité est laissée à un agent du logiciel, le système nécessite que l'agent logiciel se comporte de la manière qui lui est demandée. Ils se représentent de la manière suivante :

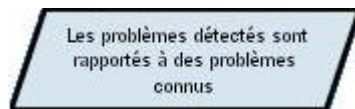


Figure 23 : Requirement

- Les *expectations* qui sont des buts pour lesquels la réalisation est confiée à un agent qui fait partie de l'environnement du système, on ne peut donc pas garantir le comportement de ces agents, mais on s'attend à ce qu'ils se comportent d'une

certaine manière. Ils se représentent de la manière suivante :



Figure 24 : Expectation

Ces buts terminaux sont affectés à des agents, la représentation de cette relation se fait à l'aide de flèches comme pour les raffinements de buts.

## 4.2 UML

**UML, langage de modélisation.** UML (« Unified Modeling Language ») est un langage de conception se basant sur la création de modèles successifs de plus en plus détaillés pour mettre en place une solution au problème étudié. UML a succédé à une série de méthodes d'analyse et de conception orienté-objet au début des années 90. Il est la fusion et la synthèse des trois méthodologies dominantes de Booch (OOT), de Rumbaugh (OMT) et de Jacobson (OOSE), bien que sa portée soit beaucoup plus vaste. En 1997, UML a subi un processus de normalisation de la part de l'OMG (« Object Management Group »), le consortium chargé de définir les standards de l'industrie, et est aujourd'hui devenu incontournable.

UML est le fruit d'un travail d'experts reconnus ; il est issu du terrain, ce qui est un atout indéniable. De plus il est riche car il couvre toutes les phases d'un cycle de développement et il est ouvert car indépendant du domaine d'application et des langages d'implémentation. Etant le résultat d'un large consensus parmi les méthodologistes et les industriels, il s'impose aujourd'hui comme le standard de l'industrie.

Enfin, il existe de nombreux outils qui supportent UML : Rational Rose d'IBM, Together de Borland, Visio Pro de Microsoft, etc.

Les différentes phases du développement UML peuvent être représentées au moyen d'une série de diagrammes permettant de comprendre de manière visuelle les concepts définis. Tous les modèles s'enchaînent en passant de l'analyse à la conception, gagnant en complexité, s'affinant au fur et à mesure pour arriver à l'élaboration finale du modèle. Les diagrammes permettent de comprendre sous différents angles la globalité du cas étudié en présentant une vue statique et dynamique de celui-ci. Chaque diagramme exprime une partie de la structure totale, tout en étant un aspect particulier du système.



Ces diagrammes sont répartis en trois catégories. Chacune de ces catégories apporte une approche différente du modèle. Les diagrammes *statiques* permettent de décrire les aspects structurels d'un système, c'est-à-dire les éléments qui le composent. Les diagrammes *dynamiques* décrivent plutôt les aspects comportementaux du système, c'est-à-dire la manière dont un système réagit en fonction de certains événements ou actions. Enfin, les diagrammes d'*implémentation* décrivent les éléments nécessaires au déploiement de ce système lors de l'implémentation.

## 5. INGENIERIE DES BESOINS

Comme précisé précédemment, l'objectif de cette itération est de fournir l'analyse et le développement d'un outil informatique fournissant à l'ingénieur de production un reporting des différents problèmes pouvant apparaître sur les fours de la cokerie. Dans un premier temps, l'application se limitera aux problèmes des fours et à la gestion de leurs entretiens. Leurs différents dysfonctionnements peuvent être diagnostiqués à l'aide d'un ensemble de données existant déjà dans la base de données de la cokerie. Comme vu ci-avant, les principales données exploitables concernent les données relatives au défournement, le schéma en figure 25 montre que celles-ci se retrouvent dans la base de données.

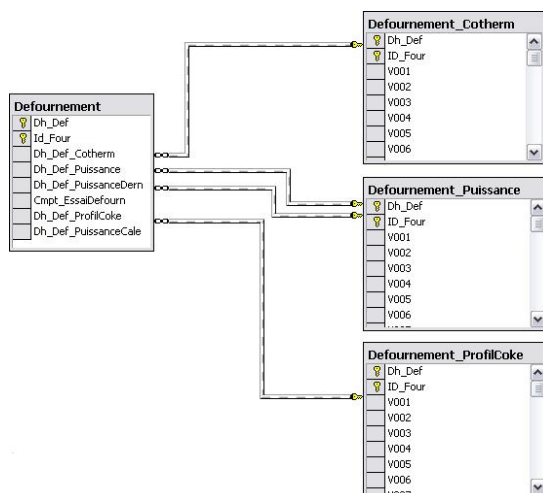


Figure 25 : Extrait du schéma de la base de données de Carsid

L'analyse de cette application a été élaborée en plusieurs phases, elle débute par un modèle des objectifs de l'application afin d'identifier intervenants et contraintes qui doivent être pris en compte par la suite. Ensuite un modèle UML

complet suivant la méthodologie définie par le UP a été développé.

### 5.1 Modélisation KAOS de la problématique four

Le diagramme de buts obtenu via l'utilisation de la méthodologie KAOS permet de définir l'ensemble des objectifs de l'application à développer et de replacer celle-ci dans son contexte d'utilisation, en identifiant les différents intervenants et leurs rôles au sein du système. Ce diagramme est présent en annexe 1, une spécification complète des buts qui y sont représentés peut-être trouvée dans [DH05]. Ce modèle des buts permet de mettre en évidence les différentes fonctionnalités attendues du système et les différentes étapes permettant d'y parvenir.

### 5.2 UML

L'ensemble des diagrammes UML effectués pour l'analyse de l'application ne sont pas présentés ici afin de ne pas alourdir la présentation. Seuls le diagramme des cas d'utilisations métier<sup>1</sup> et le diagramme de classes seront présentés ici, le lecteur désireux de consulter les autres se référera à [DH05]. Ceux-ci comprennent ;

- des diagrammes use cases montrant les différentes fonctionnalités du logiciel qui sont complétés par des diagrammes d'activités ;
- différents diagrammes de séquence et de collaboration qui permettent d'expliquer le comportement des objets de l'application.

#### 5.2.1 Le diagramme business use cases

Ce diagramme est disponible en annexe 2. Il représente l'ensemble de business use cases de la cokerie. Leur but est d'apporter de la valeur à un acteur interne ou externe à l'organisation. Plusieurs 'business goal' se distinguent dans ce diagramme, ceux-ci représentent des objectifs stratégiques de l'entreprise.

- *Avoir coke dans délai pour haut-fourneau.*  
Le but est de fournir le charbon au client. En effet celui-ci ne peut se trouver à cours de coke, les conséquences en seraient trop coûteuses pour l'entreprise : arrêt de la

<sup>1</sup> Pour respecter la terminologie anglaise, le diagramme des cas d'utilisations métier sera nommé *business use cases* dans la suite de ce travail. De même les cas d'utilisations prendront la dénomination de *use cases*.

production, coût et temps pour rendre le haut-fourneau opérationnel, etc.

Il s'agit donc d'un but stratégique de grande importance pour l'entreprise Carsid qui ne peut se permettre d'arrêter sa production ;

- *Fournir cokerie en charbon.* La réalisation de ce but ne dépend pas que de l'organisation. En effet, l'arrivée de charbon dépend de différents facteurs : respect des délais par les fournisseurs, arrivée du charbon par bateau jusqu'au port d'Anvers, transport du charbon par voie ferroviaire par la SNCB, etc.

L'entreprise dispose d'un stock de réserve de charbon pour éviter les pénuries mais des grèves prolongées ou des disfonctionnements importants pourraient mettre en péril la production de coke et par propagation de proche en proche, celle de toute la chaîne de production de l'entreprise. Il est donc essentiel que la cokerie soit bien approvisionnée et qu'un contrôle permanent soit effectué sur les stocks de charbon et sur les différents fournisseurs et intermédiaires pour parer à toute pénurie.

Il s'agit donc d'un but stratégique à l'entreprise car sans charbon, elle ne peut assurer le suivi de sa production ;

- *Défournement correctement chacun des fours.* Cette opération est essentielle pour le bon fonctionnement de la cokerie : les fours doivent être défournés au moment optimal pour respecter le compromis entre qualité du coke produit et rentabilité des fours. Un défournement peut également causer des dégâts et de ce fait ralentir la production : le four devra être réparé et sa production suspendue.

Le respect de ce but fait donc partie intégrante de la stratégie de l'entreprise qui veillera avant tout à produire la plus grande quantité de coke possible et en conséquence à réduire le temps de production.

Outre ces principaux buts stratégiques, le diagramme présente les principaux cas d'utilisation du système qui vont fournir une valeur ajoutée aux acteurs. Le diagramme modélise avant tout les différents éléments qui vont entrer en compte dans le développement de l'application et n'est pas exhaustif quant à la modélisation exacte des processus sidérurgiques d'enfournement et de défournement. De même, d'autres schémas représentant plus en détail certaines de ces activités ont déjà été modélisés dans les itérations précédentes du projet. Des diagrammes d'activités mettant en scène les différentes interactions entre acteurs du système sont présents dans le fichier [DH05].

Les use cases concernant spécifiquement la mise en place de l'application sont : '*détecter problèmes éventuels*' et '*signaler problèmes éventuels*'; l'application visant à mettre en place un outil de reporting concernant les divers événements se déroulant sur les fours. Apparaissent également sur le diagramme un ensemble de cas d'utilisation mettant en cause les divers éléments qui vont permettre de récolter l'information concernant l'état des différents fours.

## 5.2.2 Le diagramme de classes

Le diagramme de classes présenté ici ne reprend pas l'ensemble des classes du système mais uniquement les classes en relation avec l'application de gestion de fours. Il s'intègre dans le diagramme de classes complet de la cokerie disponible dans [WLK04]. Les principales classes nécessaires à l'application sont les suivantes :

### 5.2.2.1 La classe « défournement »

Etant donné que ce travail est destiné à analyser les problèmes de défournement, cette classe est l'une des principales du diagramme. Elle contient l'ensemble des attributs relatifs au défournement : l'intensité maximale atteinte par la défourneuse, le PAD, les températures moyennes, minimales, maximales, les heures d'enfournement et de défournement ainsi que l'écart type et la moyenne des températures Cotherm. La classe contient également un constructeur et des accesseurs.

En lien avec cette classe on retrouve les classes suivantes :

- La classe *disfonctionnement* servant à enregistrer les divers disfonctionnements pouvant survenir sur les fours ;
- La classe *historique*, classe d'agrégation composée de plusieurs instances de la classe *défournement*. En outre, cette classe contient diverses méthodes permettant de réaliser des comparaisons graphiques pour un attribut donné du défournement sur cet ensemble de défournements ;
- La classe *rapport\_def* est une classe destinée aux contremaîtres des équipes four. Elle a pour attributs la date de ce rapport et le libellé inscrit par le contremaître ;
- La classe *four* est la classe qui représente un four, avec pour attributs le numéro de ce four et son état, à savoir s'il est en état de fonctionnement ou non. Cette classe est reliée par une relation d'agrégation avec la classe *batterie* représentant comme son nom l'indique une batterie de fours.



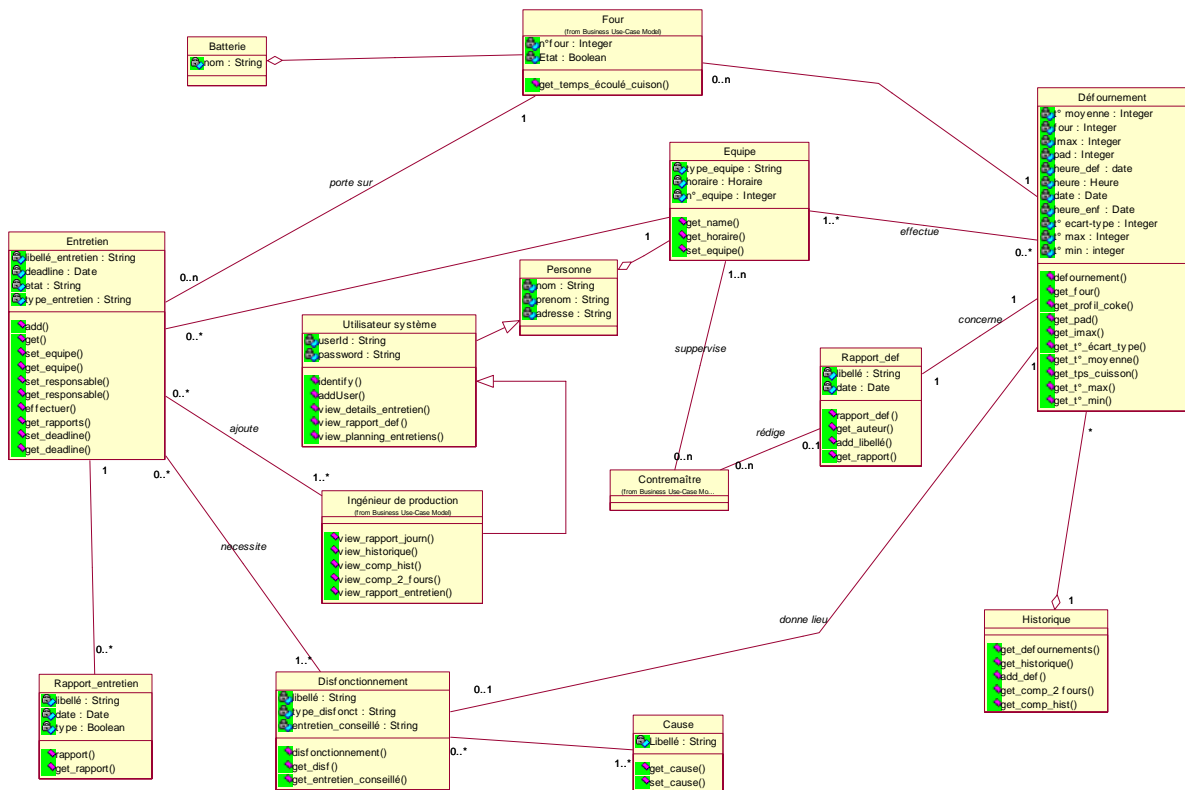


Figure 26 : Diagramme de classes

### 5.2.2.2 La classe « entretien »

Cette classe est la seconde classe centrale du diagramme, elle a comme attributs la deadline pour laquelle l'entretien doit être fait, le libellé écrit par l'ingénieur de production qui décrit en détails ce qui doit être réalisé et l'état de l'entretien qui définit si celui-ci est complètement réalisé ou non.

La classe *rapport\_entretien* regroupe les rapports d'entretiens réalisés par les équipe en charge des entretiens concernés, avec pour attributs la date de ce rapport, un libellé afin de spécifier les éventuels problèmes rencontrés et/ou causes de retards et enfin un type de rapport servant à spécifier si l'entretien a été réalisé ou non.

### 5.2.2.3 Les autres classes

Les classes restantes concernent essentiellement le personnel et sa composition :

- La classe *personne* reprend chaque travailleur de l'entreprise avec ses coordonnées pour attributs, des ensembles de personnes sont agrégés en classe *équipe* qui contiendra en plus le nom de cette équipe et son horaire ;

- La classe *utilisateur système* est une spécialisation de la classe *personne*, elle représente toute personne ayant un accès total ou partiel au système développé, c'est pourquoi elle a un mot de passe pour attribut. Cette même classe se divise en deux spécialisation : d'une part le *contremaître* qui a un accès partiel au système et d'autre part l'*ingénieur de production* qui a un accès total.

## 6. IMPLÉMENTATION

Une fois l'analyse terminée, l'implémentation de l'application qui se fonde sur elle peut-être abordée. Dans cette section, les composants nécessaires à l'implémentation seront présentés, ensuite l'implémentation et les résultats obtenus seront exposés. Enfin le déploiement des différents composants sur les installations physiques de l'entreprise sera abordé.

## 6.1 Modèle des composants

Le modèle des composants permet de définir les différents modules qui vont être implémentés ainsi que leurs interactions.

Le modèle des composants n'a pas été réalisé à l'aide du formalisme spécifié des diagrammes de composants UML mais à l'aide d'un formalisme dérivé plus riche et présent dans l'outil MS Visio 2003 ([www.microsoft.com/office/visio/](http://www.microsoft.com/office/visio/)). Ce diagramme présenté en figure 27 expose les différents composants nécessaires à la réalisation de l'application.

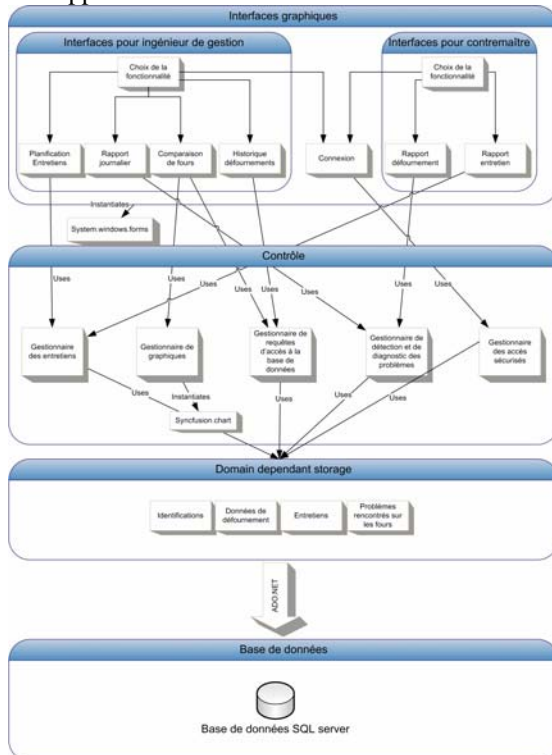


Figure 27 : Diagramme de composants

La modularité est l'un des critères prépondérants lors de la réalisation d'une architecture logicielle, c'est pourquoi il est important de séparer les différents composants à l'aide de couches qui vont permettre de limiter l'adaptation à une partie du code uniquement lors d'un changement technologique.

A cet effet, le modèle des composants de cette application compte quatre couches principales :

- La première concerne les différentes interfaces graphiques des utilisateurs ;
- La deuxième concerne tous les composants qui vont gérer les différents accès aux données ou effectuer les différentes opérations d'analyse ;

- La troisième concerne les données telles qu'elles seront manipulées par les opérations ;

- La quatrième concerne la base de données.

Le langage qui sera utilisé pour implémenter les différents composants est *Visual Basic.NET* (<http://msdn.microsoft.com/vbasic/>). Il est en effet le langage utilisé par Carsid pour toutes ces réalisations informatiques.

### 6.1.1 Couche des interfaces graphiques

L'ensemble des composants de l'interface graphique instancie *system.windows.forms*, un package permettant la réalisation de classes graphiques. L'interface graphique est réalisée à l'aide de plusieurs composants, d'abord parce que l'application offre des services différents à l'utilisateur selon que celui-ci soit ingénieur de production ou contremaître ensuite parce que les fonctionnalités sont très différentes entre-elles et nécessitent leur propre interface graphique. Il y a donc un composant graphique proposé pour chaque fonctionnalité. Le composant '*connexion*' est commun aux différents types d'utilisateurs car il offre les mêmes options à chacun.

### 6.1.2 Couche de contrôle

Les composants de contrôle sont chacun utilisés par un ou plusieurs composants de l'interface graphique, ils vont contrôler les différentes lectures ou écritures dans la base de données ainsi qu'effectuer diverses opérations. Les différents composants de contrôle sont :

- Le composant '*gestionnaire des entretiens*', qui va gérer tout ce qui concerne les entretiens, c'est lui qui empêche de prévoir deux entretiens en même temps pour la même équipe ;
- Le composant '*gestionnaire de détection et de diagnostic des problèmes*' va analyser les différentes données de défournement afin de détecter la présence de problèmes et trouver la cause de ceux-ci ;
- Le composant '*gestionnaire des accès sécurisés*' va vérifier les données introduites par l'utilisateur lors de son identification pour lui donner accès aux différents services de l'application ;
- Le composant '*gestionnaire de graphiques*' va réaliser les différents graphiques de comparaisons à l'aide du package *synchfusion.chart* ;
- Le composant '*gestionnaire des requêtes d'accès à la base de données*' permet d'obtenir des accès en lecture aux données

de défournement se trouvant sur la base de données.

### 6.1.3 Couche des données utilisées

Presque tous les composants de la couche de contrôle font appel aux différentes données se trouvant dans la couche *domain dependant storage* qui permet de représenter les données spécifiques à l'application de manière abstraite, indépendamment de la base de données physique qui va les contenir. Les différents éléments nécessaires à l'application qui y sont enregistrés sont :

- Les données relatives au défournement (puissance, cotherm, profilocoke, durée de cuisson) ;
- Les données d'identification (noms d'utilisateurs, mots de passe) ;
- Les différents entretiens planifiés ;
- Les différents problèmes rencontrés sur les fours et leurs causes.

### 6.1.4 Couche de la base de données

Ces données sont physiquement stockées sur une base de données gérée par le système de gestion de base de données, *SQL server 2000* ([www.microsoft.com/sql/](http://www.microsoft.com/sql/)), dont un serveur est déjà présent dans l'infrastructure informatique de Carsid. La transition des données entre la base de données et le *domain dependant storage* se fait à l'aide du système d'accès aux bases de données *ADO.NET*.

## 6.2 Développement

Dans cette partie vont être présentées les différentes fonctionnalités des applications qui ont été développées au terme de cette itération. Il y a trois applications, une destinée à l'ingénieur d'amélioration, une pour les contremaîtres des équipes fours et une pour les contremaîtres des équipes d'entretien. Pour chacune de ces applications, seront présentés les fonctionnalités qui les composent accompagnés de captures d'écran des fenêtres ainsi qu'une brève description de leur fonctionnement.

### 6.2.1 Les applications implémentées

#### 6.2.1.1 L'application de l'ingénieur de production

Cette application est celle qui permet à l'ingénieur de production de lancer tous les utilitaires développés dans ce travail. Une fois l'application lancée, une boîte à outils va apparaître.

L'ingénieur devra cliquer sur « Connexion » pour ouvrir une fenêtre de le login.

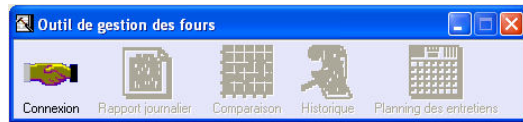


Figure 28 : La fenêtre principale avant connexion

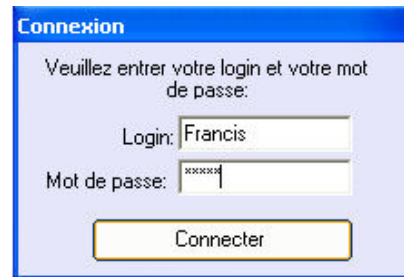


Figure 29 : La fenêtre de login

Une fois l'accès autorisé, les autres outils deviennent disponibles.

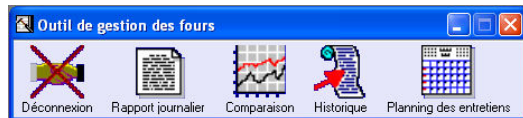


Figure 30 : La fenêtre principale après connexion

#### 6.2.1.1.1 L'outil « rapport journalier »

Cet utilitaire permet de voir le rapport journalier qui contient les rapports d'entretien et de défournement du jour. L'interface se compose de trois onglets :

- L'onglet « Défournements » contient une liste des défournements avec les incidents du jour accompagnée des données des rapports concernant ces défournements. L'utilisateur peut cliquer sur le petit sigle '+' devant chaque entrée de la liste afin d'obtenir plus de détails concernant ce défournement. En dessous de la liste se trouve un formulaire pour pouvoir ajouter un entretien suite à un problème rapporté.

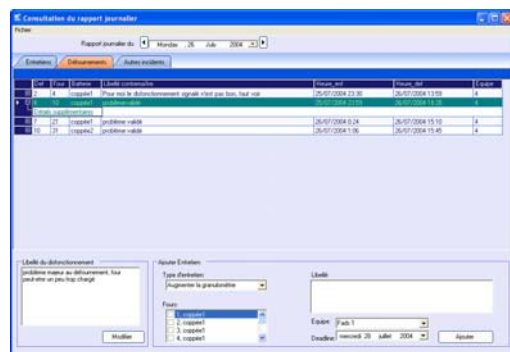


Figure 31 : L'onglet "défournements"



Figure 32 : La grille après le click sur "détails"

- L'onglet «Entretiens » contient une liste des rapports d'entretien rédigés pendant la journée avec les données de ces rapports d'entretien. Si l'utilisateur sélectionne une ligne de la liste, les données relatives à l'entretien de ce rapport seront affichées dans le formulaire situé en dessous. S'il le désire, l'ingénieur de production pourra modifier ces données.

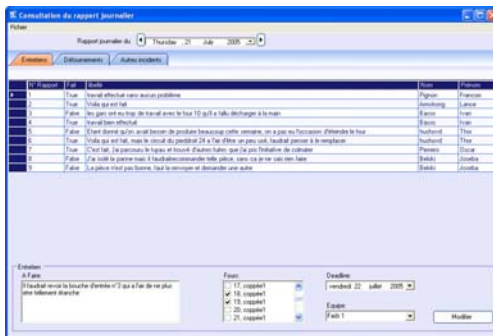


Figure 33 : L'onglet "Entretiens"

- L'onglet « autres incidents » permet de voir les autres incidents survenus dans la journée qui ne concernent pas les fours. Enfin, s'il le souhaite, l'ingénieur pourra voir les rapports journaliers des jours précédents en sélectionnant une autre date.

#### 6.2.1.1.2 L'outil « Historique »

Cet outil permet à l'ingénieur de consulter tous les défournements et entretiens enregistrés à ce jour. Ici encore, l'interface dispose de deux onglets :

- L'onglet « entretiens » affiche tous les entretiens réalisés ou non par ordre d'identifiant. Un clic sur le sigle '+' permet de voir les fours concernés par cet entretien, les causes de cet entretien et les rapports associés. Il est également possible de regrouper les entretiens en effectuant un drag-and-drop de l'attribut à utiliser pour le regroupement sur la barre située en haut de la grille.

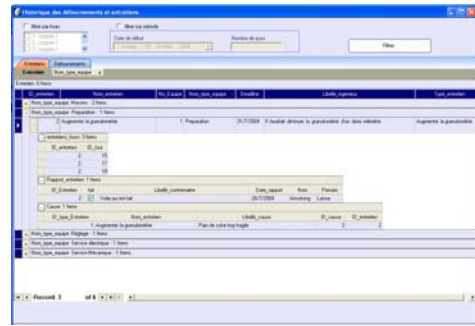


Figure 34 : "Entretiens" de l'historique

- L'onglet « Défournements » affiche tous les défournements déjà réalisés par ordre d'identifiant. Un clic sur le sigle '+' permet d'obtenir des détails supplémentaires.



Figure 35 : L'onglet "Défournements" de l'historique

De plus, l'utilisateur peut créer un filtre pour n'afficher que les données concernant certains fours et/ou une certaine période.

#### 6.2.1.1.3 L'outil « planning des entretiens »

Cet outil permet à l'ingénieur de visualiser les entretiens par « deadline » et de replanifier ceux-ci. Il peut éventuellement en supprimer et en ajouter. L'interface se compose d'un calendrier permettant à l'utilisateur de choisir une date. Un clic sur une date affichera une grille comprenant les entretiens à réaliser pour cette date. Lorsqu'une ligne de la grille est sélectionnée, les données de l'entretien s'affichent à droite du calendrier. Trois options sont alors offertes, l'utilisateur peut changer ces données et les appliquer à l'entretien sélectionné en appuyant sur « Modifier », l'appui sur le bouton « Ajouter » ajoutera un nouvel entretien avec comme paramètres les données saisies et enfin l'appui sur le bouton « Supprimer » effacera l'entretien sélectionné.

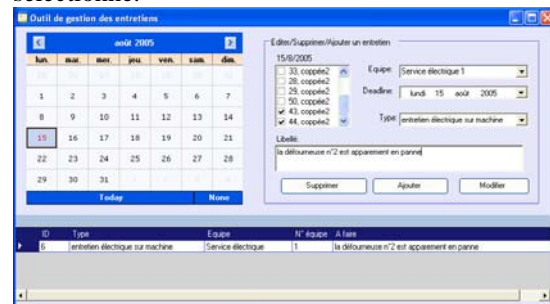


Figure 36 : L'outil de gestion des entretiens



### 6.2.1.1.4 L'outil « comparaison de défournements »

Cet outil permet de visualiser de façon graphique l'évolution des différents paramètres obtenus aux défournements afin de pouvoir déceler certaines anomalies. L'interface est constituée de deux onglets :

L'onglet « comparaison entre fours » permet de visualiser sur un même graphique, pour chaque paramètre, l'évolution de ce paramètre au cours d'une période définie pour les fours sélectionnés. La pression sur le bouton « tracer » va faire apparaître cinq onglets chacun contenant le graphe d'un paramètre.

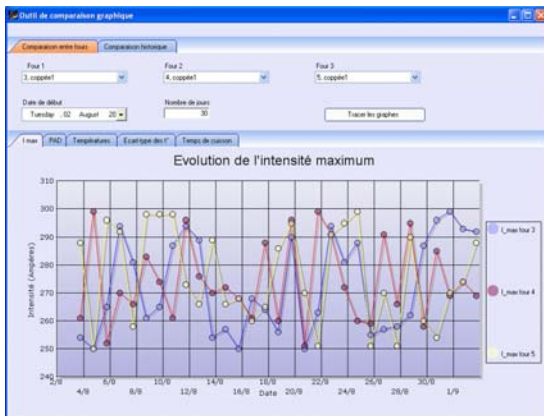


Figure 37 : Comparaison entre fours

L'onglet « comparaison historique » permet de visualiser, pour chaque paramètre, l'évolution de ce paramètre pour le four sélectionné et pour chacune des périodes sélectionnées. La pression sur le bouton « tracer » va faire apparaître cinq onglets chacun contenant un graphe pour chaque période représentant l'évolution d'un paramètre.

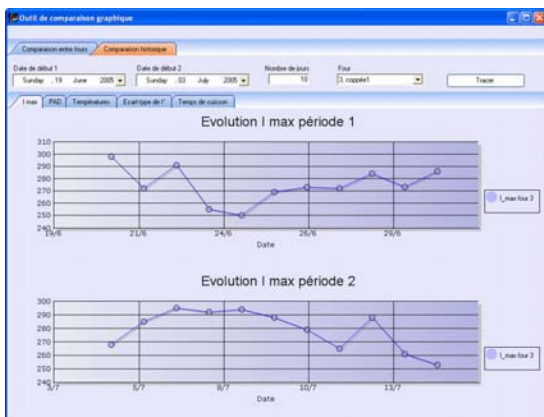


Figure 38 : Comparaison entre deux périodes

Il est également possible de zoomer sur les graphes en sélectionnant une zone de ceux-ci à l'aide de la souris.

### 6.2.1.2 Les outils destinés aux contremaîtres

#### 6.2.1.2.1 L'outil « rapport de défournement »

Cet outil destiné aux contremaîtres des équipes fours permet à ceux-ci, un fois loggés, de compléter les rapports de défournement de la journée générés par le système en cas de disfonctionnement. Ils peuvent aussi ajouter d'autres disfonctionnements ne concernant pas les fours. Si le contremaître a oublié de faire ses rapports la veille il peut les rééditer.

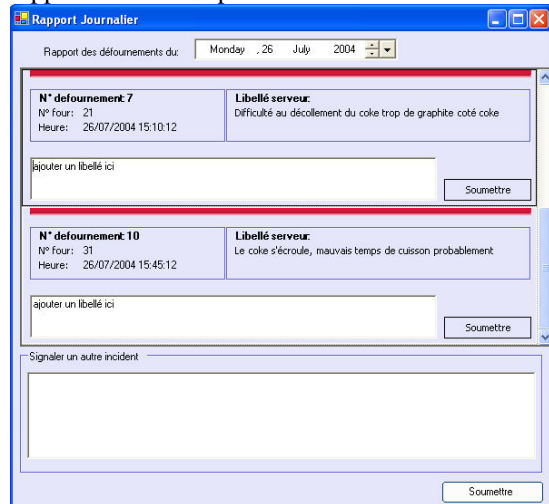


Figure 39: L'outil "Rapports de défournement"

#### 6.2.1.2.2 L'outil « Création de rapports d'entretien »

Cet outil destiné aux contremaîtres des équipes d'entretien permet à ceux-ci, une fois loggés de rédiger des rapports d'entretien. Dans la fenêtre sont affichés tous les entretiens qui n'ont pas encore été réalisés. L'utilisateur n'a plus qu'à introduire un libellé dans le champ destiné à cet effet, à cocher la case précisant si l'entretien est terminé ou non et enfin à soumettre le rapport.



Figure 40 : L'outil "Rapports d'entretien"

## 6.3 Déploiement

L'implémentation des divers composants terminée, il faut replacer ceux-ci dans le cadre de l'infrastructure informatique de Carsid. Le diagramme de déploiement présenté en figure 41 permet de présenter les éléments physiques du système et la localisation des différents composants sur cette infrastructure à l'aide des principaux éléments de ce diagramme les *processors* (machines du système qui disposent d'un processeur) et les *devices* (matériel physique du système qui ne dispose pas de processeur).

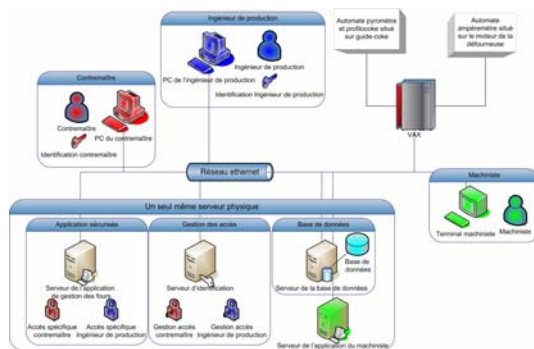


Figure 41 : Diagramme de déploiement

A l'instar du modèle de composants, le modèle de déploiement n'a pas été réalisé à l'aide du formalisme spécifié des diagrammes de composants UML mais à l'aide d'un formalisme dérivé plus riche et présent dans l'outil MS Visio 2003.

### 6.3.1 Processors

Les principaux processors de l'infrastructure sont les suivants :

- Le serveur sur lequel tourne la base de données et les applications. Il est ici séparé en plusieurs entités logiques qui vont permettre de séparer les composants. Il n'est pas utile vu l'ampleur du système mis en place de séparer physiquement ces serveurs, le nombre d'utilisateurs, l'importance des accès et la vitesse à laquelle sont récoltées les données ne justifie pas une protection coûteuse de gestion de la concurrence des requêtes.

Il comprend :

- Une entité qui gère la base de données, à savoir les différents accès en lecture et en écriture sur celle-ci. Elle est gérée par l'outil Microsoft SQL Server 2000 ([www.microsoft.com/sql/](http://www.microsoft.com/sql/)). L'accès à ce serveur ne se fait pas directement

mais par l'intermédiaire d'applications qui donnent l'information aux utilisateurs sous une forme moins brute ;

- Une autre entité qui gère les accès des utilisateurs à l'application : le serveur d'identification. Celui-ci gère les différents accès des utilisateurs afin de garantir que chacun d'entre eux n'a accès qu'aux fonctionnalités de l'application qui sont développées pour lui et qu'une personne étrangère au système n'aie aucun accès à l'application. Les personnes devant avoir accès au système disposent toutes d'une identification le leur permettant ;
- Une entité gérant l'application développée : le serveur d'application. L'accès à celui-ci est restreint aux personnes s'étant préalablement identifiées sur le serveur d'identification ;
- Un dernier serveur gérant l'application du machiniste qui s'occupe des défournements et du réglage des fours. Cette application vise uniquement des consultations des données de défournement.
- L'ordinateur du contremaître, via lequel il se connecte à l'application. Il doit préalablement entrer son nom d'utilisateur et son mot de passe pour avoir accès à l'application ;
- L'ordinateur de l'ingénieur de production, duquel celui-ci se connecte à l'application. Il doit également s'identifier pour avoir accès aux différentes applications qui l'intéresse ;
- Le système Vax (<http://h18002.www1.hp.com/alphaserver/vax/>) qui permet de traduire les données provenant des différents automates capturant les données de défournement et de les mettre sur le réseau pour qu'ils soient enregistrés dans la base de données.

### 6.3.2 Devices

Les différents devices intervenant sont :

- Le réseau ethernet sur lequel transitent toutes les données ;
- Les différents automates récoltant les données de défournement, qu'ils soient situés sur le moteur de la défourneuse ou sur le guide coke. Le fonctionnement de ceux-ci



est simple, un automate Tsx<sup>2</sup> fournit un signal à un codeur Satec<sup>3</sup> qui envoie les données sous formes d'ondes jusqu'à un décodeur Satec distant via les rails des installations. Le signal sortant du décodeur est récupéré par un automate Tsx qui transmet ces données à une plate-forme Vax via le réseau ;

- Le terminal du machiniste qui lui permet d'accéder aux schémas et courbes de défournements susceptibles de l'aider à régler ses fours.

### 6.3.3 Localisation des composants

Les différents composants sont installés comme suit sur les différents processors :

- Serveur de l'application de gestion des fours : l'ensemble des composants graphiques, les différents composants de contrôle (gestionnaire des entretiens, gestionnaire de détection et de diagnostic des problèmes, gestionnaire de graphiques et le gestionnaire des requêtes d'accès à la base de données) et les composants du *domain dependant storage* sont installées sur ce serveur ;
- Serveur d'identification : seul le composant 'gestionnaire des accès sécurisés' est installé sur cette machine ;
- Serveur de la base de données : le composant base de données est bien évidemment localisé sur ce processor ;
- Serveur de l'application du machiniste : ne concerne pas les composants créés pour la nouvelle application.

Aucun composant n'est installé sur l'ordinateur de l'ingénieur de production ou sur celui du contremaître.

## 7. CONCLUSION

La quatrième étape du projet de mise à jour des systèmes d'information de la cokerie de Carsid, présentée dans ce travail s'achève ici. Centrée sur

---

<sup>2</sup> les automates Tsx sont des automates industriels programmables qui interprètent des données en entrée et, selon les valeurs, retournent une certaine sortie, par exemple, pour l'automate servant à la mesure du Cothem, le voltage en sortie est différent selon la température observée

<sup>3</sup> Equipement de transmission des données, il s'agit d'un système de codeur/décodeur avec transmission sur rails. <http://www.satec-electronique.fr/>

le reporting des problèmes au niveau des fours, elle est le fruit d'un travail commencé il y a plus de quatre ans. Cette étape enrichit les efforts précédents par l'apport d'une application permettant de gérer les fours de la cokerie et les divers problèmes pouvant s'y produire. La réalisation de cette application a nécessité un travail conséquent de modélisation inscrit dans la continuité des travaux précédents et qui lui permet une intégration optimale dans son environnement : la cokerie de Carsid.

Il est utile de résumer rapidement la démarche entreprise lors de cette étape du projet. L'analyse commence par une recherche des buts liés à la mise en place de l'application et se poursuit par une modélisation UML des cas d'utilisation métier propres à l'environnement du problème étudié. Obtenus grâce au modèle des buts, ces derniers sont complétés par des diagrammes d'activités permettant de représenter en détails leur dynamique. Vient ensuite le développement d'un modèle objet composé d'un diagramme de classe détaillé par des diagrammes de séquence, de collaboration qui permet la réalisation des buts et des cas d'utilisation identifiés dans le cadre du paradigme orienté-objet. Ce modèle et particulièrement les classes spécifiques à ce nouvel effort de modélisation s'intègrent dans le diagramme de classes réalisé lors des étapes précédentes du projet. Par la suite, un diagramme de composants expose les différents éléments qui vont permettre la réalisation de l'application. Ces composants ont ensuite été implémentés et finalement présentés dans un diagramme de déploiement qui va montrer comment ceux-ci seront intégrés dans l'infrastructure physique de la cokerie.

Le projet Carsid ne s'achève cependant pas ici, bien qu'un grand pas en avant ait été fait lors de cette étape. Il se dégage désormais en effet une application concrète visant à améliorer l'exploitation des données de la cokerie. Cette application souffre toutefois encore de quelques lacunes : elle n'est pas totalement implémentée et son efficacité en termes de diagnostic des différents problèmes apparaissant sur les fours pourrait être améliorée grâce à l'intégration d'outils d'analyse de données. Il faudrait pour cela que le personnel de la cokerie fournisse un historique complet des pannes rencontrées et des données relevées lors de ces pannes. L'utilisation de systèmes Multi-Agents pourrait, à ce niveau, s'avérer d'une grande pertinence. Il reste également d'autres domaines à couvrir dans la cokerie, la modélisation de tous les aspects n'est pas encore complète et laisse la porte ouverte à la création d'autres applications permettant d'améliorer sa gestion.

## 8. REFERENCES

- [Alh03] Alhir Si S., « *Learning UML* », O'Reilly, 2003.
- [AvL04] van Lamsweerde A., « Génie Logiciel : Méthodes de développement », 2004.
- [BB02] Boggs W., Boggs M., « *Mastering UML with Rational Rose 2002* », Sybex Inc., 2002.
- [Boe96] Boehm B., « *Anchoring the Software Process* », IEEE Software, Volume 13, No 4, July 1996, pp. 73-82.
- [BGG+96] Brun B., Gaillet J.P., Griffay G. Roth J.L., « *Modélisation thermique globale des fours à coke* », Revue de métallurgie - CIT, 1996.
- [Can92] Canon T., « Analyse graphique prévisionnelle des dégradations des parois réfractaires de fours à coke et des irrégularités de marche », Cokerie de Marchienne, 1992.
- [Can95] Canon T., « *Régularités d'exploitation* », Cokerie de Marchienne, 1995.
- [CB96] Canon T., Bury G., « *L'analyse de la puissance au défournement* », Cokerie de Marchienne, Congrès ATS-Sidmar, 1996.
- [Coc00] Cockburn A., « *Writing effective use cases* », Addison Wesley, 2000.
- [DH05] Deleval T., Herrensens C., « Analyse et développement d'une application de gestion de fours à coke : Application à la cokerie de Carsid », mémoire UCL, Lovain-la-Neuve, 2005.
- [Dew99] Dewalt C., « *Business process modelling with UML* », John Hopkins University, 1999.
- [DFK+03] Donnay A., Fouss F., Kolp M., Massart D., Pirotte A., « *Analyse orientée objet de processus sidérurgiques de type cokier* », Working paper IAG 86/03, 2003.
- [Don02] Donnay de Casteau A., « Modélisation orientée objet en UML de processus sidérurgique : Application à la cokerie de Carsid », mémoire UCL, Louvain-la-Neuve, 2002.
- [EN04] Elsmari R., Vavathe S., « *Fundamentals of database systems : Fourth edition* », Addison Wesley, 2004.
- [EP00] Eriksson H.E., Penker M., « *Business Modeling with UML : Business patterns at work* », John Wiley and sons, 2000.
- [FIK+03] Fouss F., Ibarz M., Kolp M., Pirotte A., « *Steel Production Data Warehouse Reengineering* », Working Paper IAG 85/03, 2003.
- [Fiv04] Fivez F., « *Process Cokerie* », Cokerie de Marchienne, 2004.
- [Fow03] Fowler M., « *UML Distilled* », Addison Wesley, 2003.
- [Gru02] Grundgeiger D., « *Programming Visual Basic .NET* », O'Reilly, 2002.
- [Hen01] Henmann J., « Introduction to Business modeling using the Unified Modeling Language », the Rational Edge, 2001.
- [Iba03] Ibarz Roger M., « Object Oriented reengineering of the steel product databases at Carsid », mémoire UCL, Louvain-la-Neuve, 2003.
- [Isl91] Isler D., « Comportement du coke au défournement : Approche théorique, simulation à l'échelle pilote, mesures et suivis industriels », Revue de Métallurgie – CIT, 1991.
- [JB00] Jacobson I., Bylund S., « *The Road to the Unified Software Development Process* », Cambridge University Press, 2000.
- [JBR99] Jacobson I., Booch G., Rumbaugh J., « *The Unified Software Development Process* », Addison-Wesley, 1999.
- [Kru03] Kruchten P., « *The Rational Unified Process : An Introduction* », Longman (Wokingham), Addison-Wesley, December 2003.
- [LOU04] Louvigny L., « Modélisation orientée objet d'aspects comportementaux de base de données, application à la cokerie de Carsid », mémoire UCL, Louvain-la-Neuve, 2004.
- [Lor02] Lorge G., « Synthèse des applications informatiques à la cokerie de Charleroi », Cokerie de Marchienne, 2002.
- [Mod95] Modolo F., « *Gestion des pannes et arrêts* », Cokerie de Marchienne, 1995.
- [Nil01] Nilsson J., « *.NET enterprise design with Visual Basic .NET and SQLServer 2000* », Sam's Publishing, 2001.

[Qua99] Quatrani T., « Visual Modeling with Rational Rose 2000 and UML », Addison Wesley, 1999.

[RJB98] Rumbaugh J., Jacobson I., Booch G., « *The Unified Modeling Language reference manual* », Addison Wesley, 1998.

[RUP03] IBM, « The Rational Unified Process. Version 2003.06.00.65 », Rational Software Corporation, 2003.

[Wau03] Wautelet Y., « Application de la méthode RUP/UML à l'entreprise sidérurgique Carsid », mémoire UCL, Louvain-la Neuve, 2003.

[WLK04] Wautelet Y., Louvigny L., Kolp M., « Le Unified Process comme méthodologie de gestion de projet informatique. Eléments d'application en milieu sidérurgique », IAG WP 109/04.

[WLK04b] Wautelet Y., Louvigny L., Kolp M., « Modélisation oriente-objet d'aspects opérationnels de base de données sidérurgique », IAG WP 111/04.

Sources provenant de sites Internet :

Brain M., « *How iron and steel work* », <http://science.howstuffworks.com>

« *The steel making process* », <http://www.arcelor.com>

« Comment fabrique t'on l'acier ? Un processus de transformation en trois étapes », <http://www.ffacier.org>

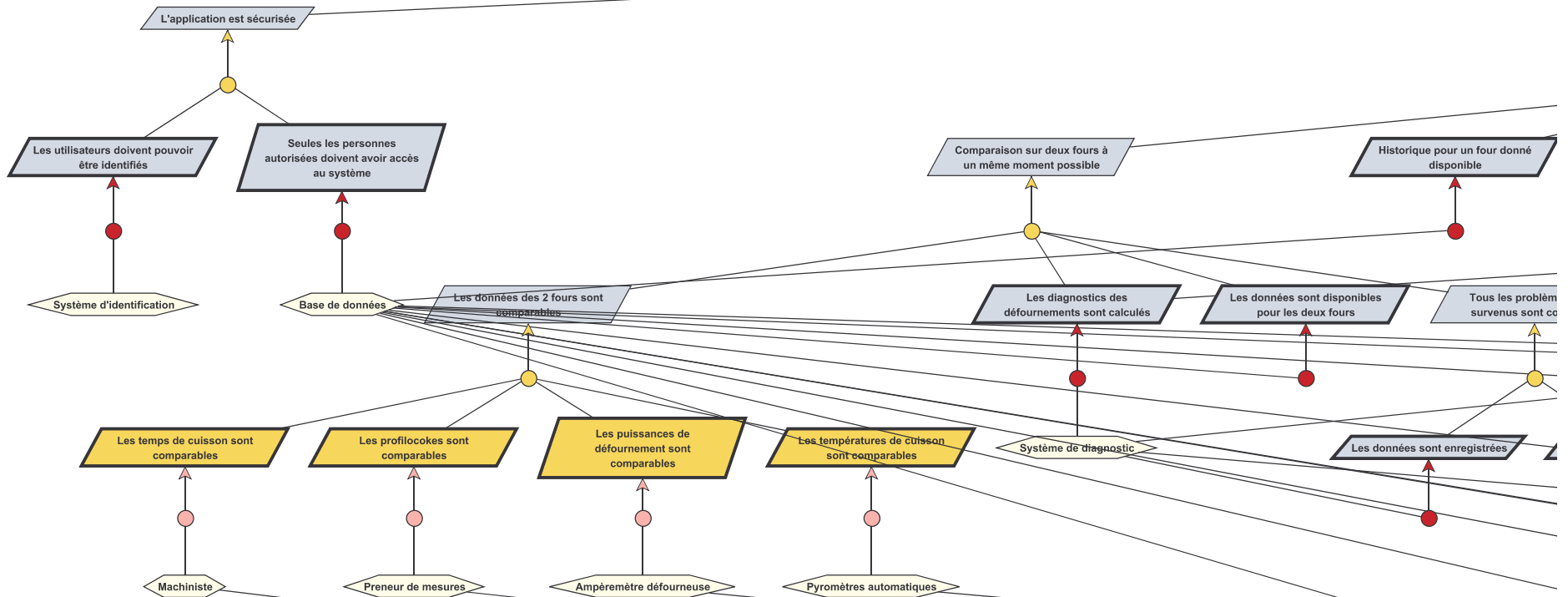
« *Production process* », <http://www.sidmar.be>

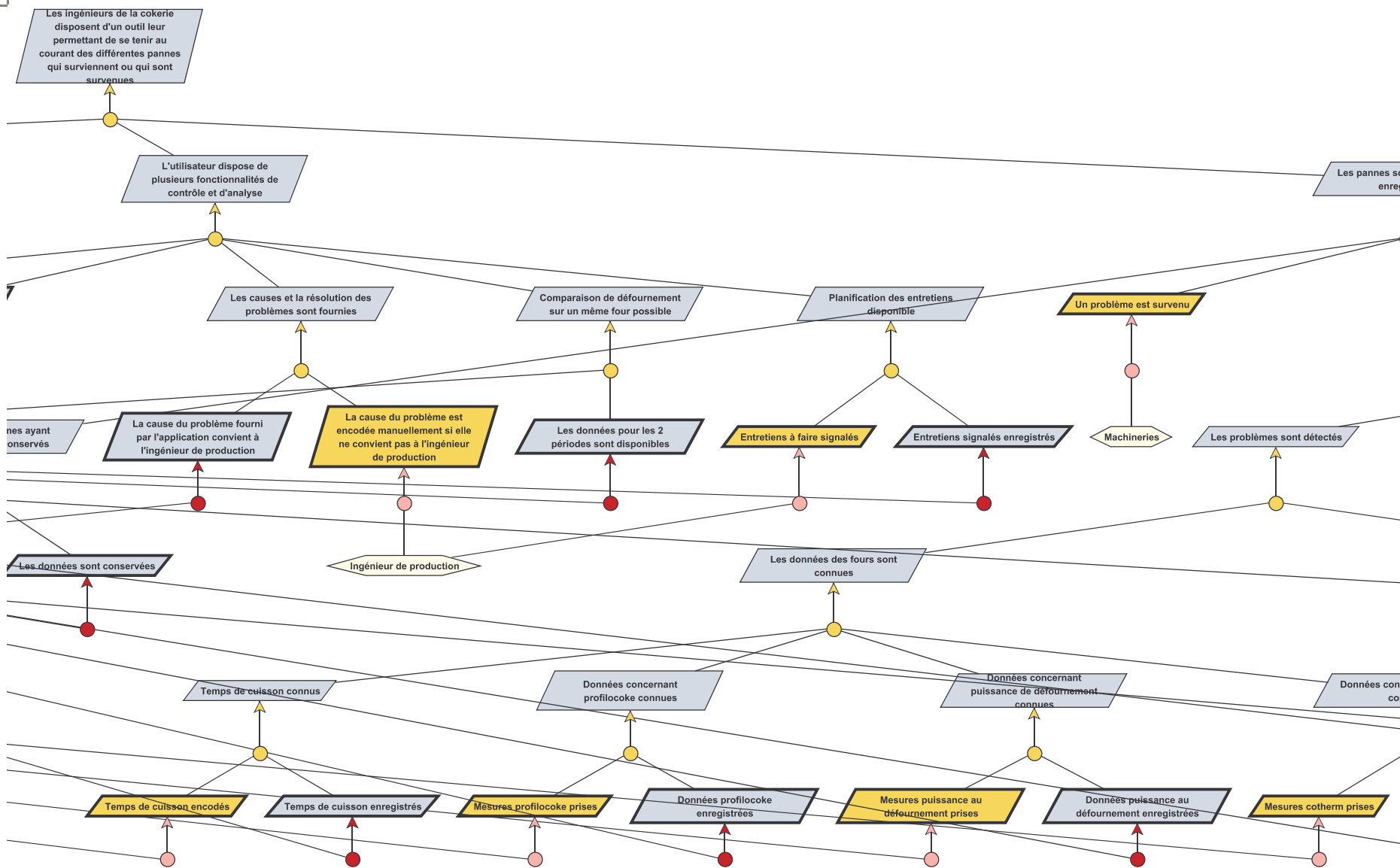
[www.duferco.com](http://www.duferco.com)

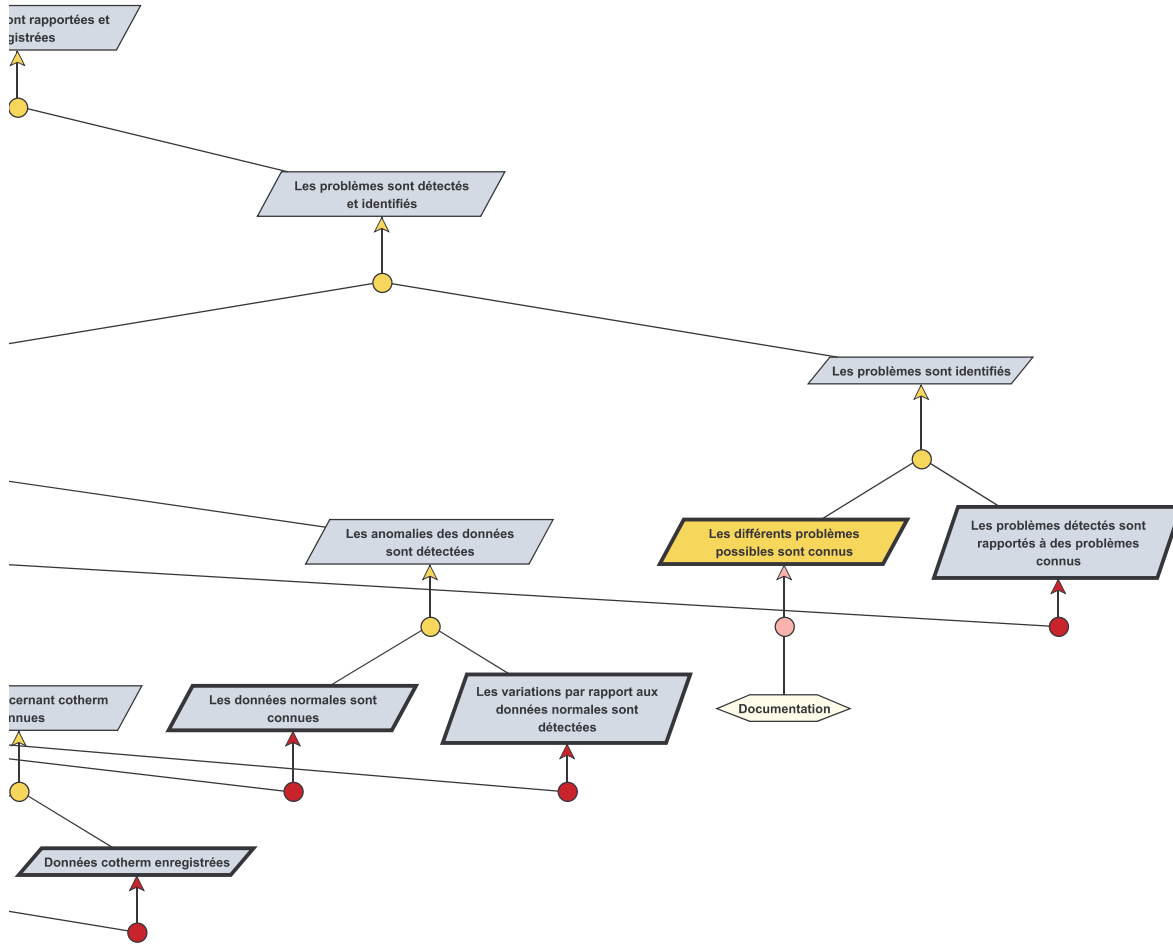
[www.arcelor.com](http://www.arcelor.com)

[www.cockerill-sambre.com](http://www.cockerill-sambre.com)

## **9. ANNEXE 1 : MODELISATION KAOS DE LA PROBLEMATIQUE FOUR**









## 10. ANNEXE 2 : DIAGRAMME DES CAS D'UTILISATION METIER

