



**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS DE MOSTAGANEM**

**Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique**

**MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : Ingénierie des Systèmes d'Information**

LA PERSONNALISATION DE LA SURVEILLANCE DES PROCESSUS METIERS

Etudiants :

**SADJI ABDELDJALIL OUSSAMA
TOUIL MOHAMED**

Encadrant(e) :

MOUSSA Mohammed

Année Universitaire 2014/2015

INTRODUCTION GÉNÉRALE

Un système qui a démontré de réels avantages à l'industrie depuis les années 1990 est *Workflow Management System* (WFMS). Un WFMS est principalement un logiciel qui est chargé de la gestion d'une exécution d'un flux de travail en interprétant la définition d'un processus, identifier les ressources impliquées, en interaction avec les participants du flux de travail, et en invoquant des applications externes si nécessaire.

Il est important de préciser que le Workflow est une partie de ce qui est connu sous le nom de *Business Process Management* (BPM). La technologie BPM implique la création, l'intégration, l'automatisation et l'optimisation des processus à l'intérieur et à l'extérieur d'une entreprise. L'une des phases les plus importantes de la gestion des processus métiers est le suivi des processus parce que les entreprises peuvent non seulement utiliser des indicateurs de processus, mais aussi des outils de suivi pour avoir une gestion efficace des ressources.

Malgré l'importance du suivi des processus métiers et les efforts qui ont déjà été mises en œuvre dans l'industrie telle que le modèle de référence proposé par la Coalition WFM, il est vrai non seulement que le marché est composé de différents types de clients, mais aussi que chaque client a différents besoins. Afin de remédier à cette situation, nous allons utiliser le terme "personnalisation", qui, dans ce contexte, est défini comme la capacité des clients de personnaliser la manière dont les fournisseurs exécutent un processus.

Il est nécessaire de préciser que la personnalisation implique une grande diversité d'aspects tels que la personnalisation de contrôle de processus, où les clients choisissent les activités qui seront effectuées, ou la personnalisation de la QoS (Quality Of Service) où le client définit le niveau de sécurité sur les ressources, les transactions ou les processus, etc.

L'aspect de la gestion est l'un des domaines où il est possible de trouver un écart parce BPM et la personnalisation n'ont pas été étudiés par la littérature, et c'est pour cette raison la question «est-il possible de trouver un modèle personnalisable générique pour la surveillance d'un processus d'affaires ? » devient si important pour notre futur travail.

L'objectif de notre travail est l'étude des processus métiers en général, et la personnalisation de ces derniers en particulier. Dans un premier temps, il s'agit de cerner les concepts et technologies permettant la modélisation et la surveillance des processus, puis voir les cadres conceptuels pour la personnalisation des processus métiers.

Ce document est structuré comme suit :

D'abord, nous introduisons le concept de la surveillance des processus métiers, ensuite nous présentons le modèle multidimensionnel pour la surveillance des processus métier, et finalement nous fournissons une architecture détaillée de la solution.

Introduction

La démarche « processus » ne date pas d'aujourd'hui, mais ces dernières années, cette approche s'est imposée peu à peu comme une démarche indispensable à la gouvernance de l'entreprise et du SI. Les stratégies d'optimisation des SI doivent viser une plus grande agilité pour réagir aux transformations et aux évolutions dictées par le marché, par la réglementation, et par l'évolution des effectifs. Connaître et maîtriser ses processus devient alors une nécessité pour appréhender au mieux les impacts des transformations, faire évoluer ou corriger rapidement et à moindre coût les pratiques de l'entreprise.

I. 1. Gestion des Processus Métier (BPM)

Le Business Process Management (BPM) apporte une traçabilité et un pilotage des processus de l'entreprise, mais également un niveau de mesure nécessaire à l'identification de problèmes potentiels au sein d'un ou plusieurs processus.

Un BPM répond aux 3 objectifs majeurs suivants :

- ✓ Il offre une qualité pour aligner la demande des clients et partenaires aux produits ou services livrés.
- ✓ Il réduit le délai pour la mise sur le marché de nouveaux produits ou services, pour la livraison de ceux-ci.
- ✓ Il offre une maîtrise des coûts afin d'être performant au niveau de ses revenus, voire efficient vis-à-vis des actionnaires.

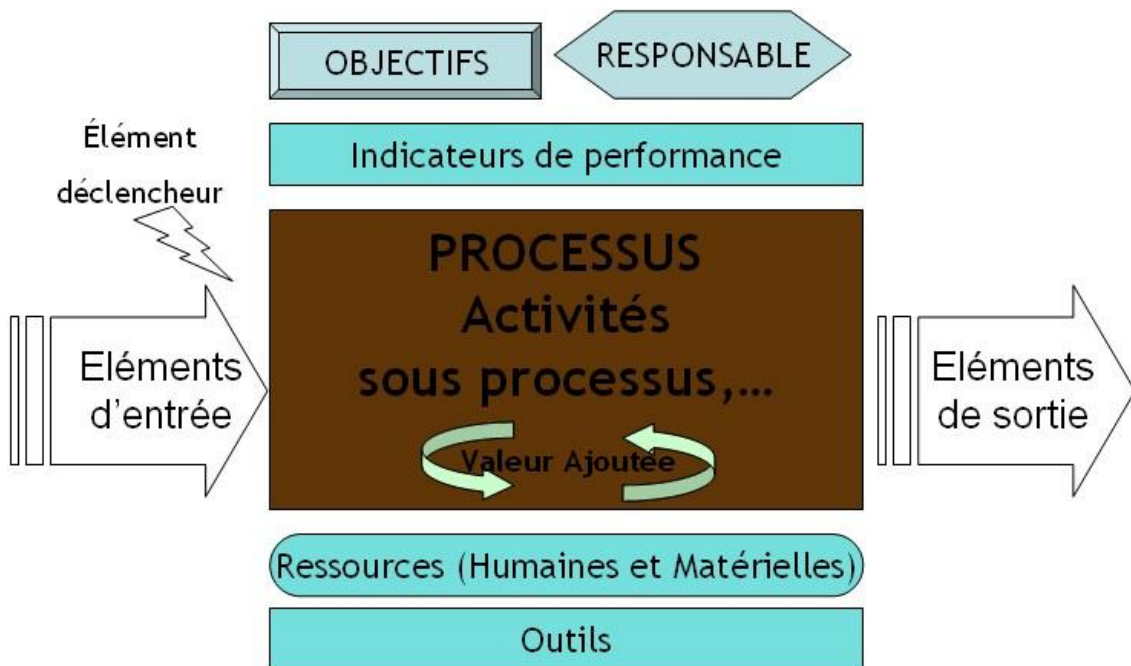


Figure I.1. Structure d'un processus

I. 1.1. Définition d'un processus métier

Un processus est un ensemble d'activités organisées dans le temps produisant un résultat précis et mesurable. En d'autres termes, un processus métier est un ensemble de tâches que des humains (le personnel de l'entreprise) ou des machines doivent réaliser pour créer de la valeur pour le client ou l'entreprise.

Dans la Figure I.1, Un processus métier est un système d'activités qui utilise des ressources pour transformer des éléments d'entrée, en éléments de sortie. Il sert des objectifs mesurables. Il possède un responsable/Pilote qui s'assure que le processus fournit les résultats attendus par rapport aux objectifs assignés

I. 1.2. Définition de BPM

Le BPM (Business Process Management) qui peut être traduit en français par « Gestion des processus métier » est un ensemble d'outils pour concevoir, exécuter et piloter les processus métiers d'une entreprise. Dans son approche la plus aboutie, elle permet de modéliser aussi bien les aspects techniques qu'humains.

I. 1.3. Le BPM sous un angle technique

Le BPM est une évolution et une standardisation des moteurs de workflows. Actuellement, les solutions BPM propriétaires et open source utilisent massivement les standards libres qui sont largement utilisés et testés. Contrairement aux anciennes méthodes qui repartent de zéro pour leur conception ou adaptaient d'anciennes technologies propriétaires.

Aujourd'hui, une solution BPM complète utilise généralement les composants suivants :

- un moteur BPEL pour l'exécution des processus
- un serveur web utilisant massivement Ajax pour :
 - Les interfaces utilisateurs
 - La présentation des indicateurs de performance.
- Une interface de modélisation pour :
 - Modéliser les processus
 - Définir les règles
 - Définir les indicateurs de performance et les tableaux de bord
 - Conception des interfaces utilisateurs.
 - Un moteur de règles
 - Un moteur de simulation
 - Un moteur d'analyse
 - Un référentiel de métadonnées.

I. 2. La surveillance des processus métiers

Quand nous parlons de la surveillance, il est nécessaire de clarifier quelques concepts importants comme : «La mesure de processus", " La surveillance de processus" et" "Business Activity Monitoring". Ces termes sont définis comme suit :

- **Mesure de processus** : est défini comme l'action pour déterminer l'exécution du processus métier en utilisant un ensemble fourni d'indicateurs d'exécution.
- **Surveillance de processus** : est défini comme l'action pour jeter un coup d'œil soigneux au-dessus d'une situation spécifique pour indiquer quelque chose à son sujet.
- **Business Activity Monitoring** : Selon le groupe de Gartner, le BAM est défini comme suit :

BAM est la surveillance, l'analyse et l'alerte en temps réel d'un nombre très significatif d'événements, accomplis en recueillant des données, des indicateurs clés de performance et d'événements à partir de plusieurs applications hétérogènes »

I. 2.1. Business Activity Monitoring (BAM)

Le terme de BAM – Business Activity Monitoring – a été introduit par le Gartner Group en 2002. Sa définition originale couvre les produits dont le but est « de fournir un accès à des indicateurs de performance métiers en temps réel pour améliorer l'efficacité des opérations métiers ». Le problème est alors de délivrer les bonnes informations aux bonnes personnes.

Littéralement, «Business Activity Monitoring » peut être traduit en « supervision des activités métiers ». Plus que de la consolidation et de la présentation d'information, on rentre dans des considérations de contrôle et de pilotage.

Le BAM doit donc permettre de délivrer les bonnes informations aux bonnes personnes, mais aussi, et surtout de déclencher des actions adéquates en fonction de la valeur d'indicateurs métiers de performance.

I. 2.2. Surveillance des processus

La surveillance des processus est fondamentale. Un des buts majeurs du BPM est de permettre un contrôle permanent et une amélioration constante des processus. Les processus sont suivis en temps réel. Lorsqu'une erreur ou une exception se produit, une alerte est envoyée par l'outil de surveillance aux utilisateurs concernés, ce qui permet une meilleure communication et une amélioration facilitée des processus.

I. 2.3. Business Process Métriques :

Une métrique est un moyen permettant de connaître la distance entre deux points. D'autre part, il est important de bien retenir que pour rendre compte d'un état, il est nécessaire de disposer de plusieurs mesures.

Voici quelques exemples de métriques métiers :

- « Combien de dossiers sont en cours de traitement ? »
- « En combien de temps en moyenne est traité un dossier aujourd'hui ? »

D'autres, plus techniques :

- « En combien de temps cette tâche est-elle exécutée ? »
- « Combien de processus sont exécutés sans erreur ? »

I. 3. Le cycle de vie de BPM

L'architecture de surveillance de base qui supporte généralement la surveillance des processus métier est composée de deux modules principaux.

- ✓ La première est le Domaine de Client et contient le module de Client de Surveillance qui est essentiellement l'interface avec les utilisateurs.
- ✓ Le deuxième est appelé Domaines de Business Process Fournisseur qui **contient** deux modules principaux Surveillance Infrastructure et Moteur de Business Process.

Le module de surveillance de l'infrastructure est généralement appelé : l'observateur, sa fonction principale est de recueillir des renseignements pertinents sur ce dernier module. Il est nécessaire de préciser que, le module de «l'infrastructure de surveillance » peut être construit soit directement au-dessus, ou dans certains cas à l'intérieur du "Moteur de Business Process".

Avec cette notion, il est possible d'avoir une meilleure compréhension de la source d'information et la composante de surveillance qui sera chargé de fournir des informations pertinentes pour l'utilisateur.

Une fois que nous avons une idée claire de l'architecture de base de surveillance, il est possible de voir le suivi du cycle de vie des processus métier de base qui est décrit ci-dessous

1. Le client de surveillance déclenche la surveillance.
2. L'infrastructure de surveillance reçoit et identifie du genre de demande du client.
3. L'infrastructure de surveillance obtient les données requises par le moteur de Business Process.
4. L'infrastructure de surveillance gère les données.

5. L'infrastructure de surveillance transfère les résultats de la demande au client de surveillance.

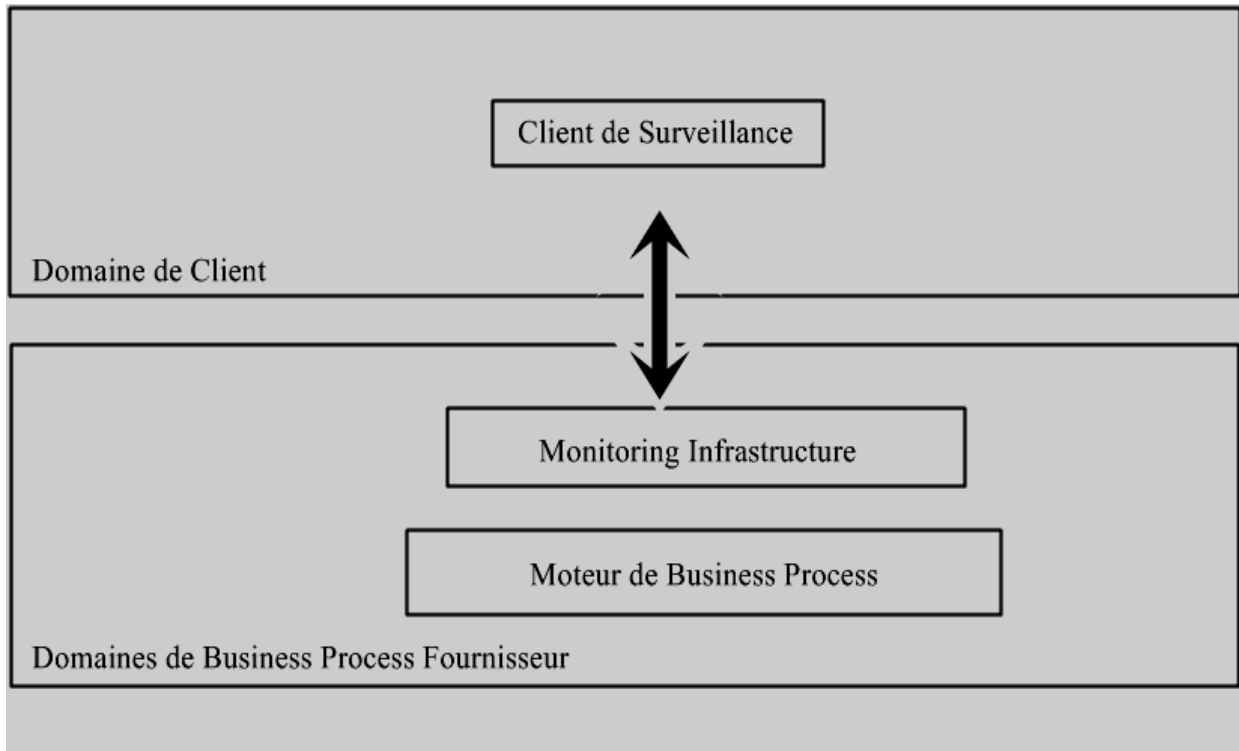


Figure I.2. L'architecture générale d'un BPM

I. 3.1. Variable de surveillance MV (Monitoring Variable)

Lorsque nous parlons de termes relatifs aux surveillances des processus métier, une variable de surveillance (MV) peut être utilisée pour faire référence à plusieurs types de données.

Par exemple, des données au niveau de l'infrastructure telle que le temps de réponse, des données au niveau de l'application, telle que le temps de l'état ou de la transformation d'une activité dans un processus spécifique [1].

La relation entre le terme MV et le cycle de vie de surveillance est essentielle. Le MV spécifie les informations de processus que l'infrastructure de surveillance obtient du moteur des processus métier (Business Process Engine) et les produits au client de surveillance.

En plus, même si le cycle de vie qui a été présenté peut être compris d'une manière simple il ya quelques détails qui doivent être souligné

- La gestion des données obtenues à partir du moteur de processus métier devrait envisager de fournir soit l'option de réécrire les données obtenues pour une MV

spécifique, soit l'option de stocker les données afin d'afficher non seulement une image capturée de la situation actuelle, mais aussi une perspective historique qui pourrait aider les utilisateurs à révéler des informations pertinentes cachées. Outre, quand on parle de contenu, il est nécessaire d'envisager d'autres options telles que le temps de disponibilité des données.

Cela signifie que l'utilisateur doit être en mesure de décider aussi si les données sont "consommées", ce qui signifie qu'elles ne seront pas disponibles dans le futur, ou «lire» ce qui signifie qu'elles seraient disponibles la prochaine fois que l'utilisateur demandera les informations sur une MV spécifique.

- Le transfert des résultats de la demande au client de pilotage est une phase qui devrait considérer les options de la façon dont les données demandées se passent depuis le service qui traite la demande, vers le client.

Les deux approches principales qui sont prises en compte ici sont "Information push" et "Information pull". Le système de pilotage devrait permettre à l'utilisateur de choisir quel genre d'options serait mieux pour la situation.

Dans cette section nous avons expliqué l'architecture élémentaire du cycle de vie de BPM, identifiant certaines options qui devraient être considérées par le modèle

I. 3.2. Les variables de BPM

Nous avons recueilli des informations à la fois sur la base de la surveillance des processus de métier et les mesures pertinentes de base qui peuvent être obtenues à partir d'un système BPM. Il est possible de voir quatre éléments principaux : les temps de données, l'état de données, les ressources de données et les niveaux d'information.

Les données des ressources ne se trouvent que dans la référence liée à un système BPM spécifique. C'est important, car cela signifie qu'il n'y avait pas clairement référence à l'information des ressources dans les sources relatives aux systèmes de BPM génériques.

Puisque le but est de développer un modèle de surveillance personnalisable générique qui peut être appliqué à toutes les BPMS, il faut que nous concentrons sur les mesures les plus susceptibles d'être présentes dans tous les BPMS afin d'assurer le succès de notre modèle. Prenant cet argument en compte, nous voyons que le temps des données, l'état des données et les niveaux d'information sont les indicateurs les plus pertinents pour nous, car ils peuvent assurer l'applicabilité de notre modèle au moins d'un point de vue théorique.

Jusqu'à présent, nous avons à la fois une vision claire de ce que les métriques devraient être considérées à partir d'un BPMS et une définition solide du cycle de vie de BPM. Cependant, nous n'avons pas expliqué comment ces deux aspects sont liés.

Dans la section précédente, la notion de la MV a été décrite et c'est la clé pour trouver le lien entre les mesures et le cycle de vie de BPM. Il est nécessaire de noter que le MV est ce qu'on doit surveiller et il est important aussi de savoir qu'une MV peut être contrôlée en fonction d'un ensemble de paramètres sont appelés «options» et les résultats varient en fonction des options sélectionnées. Cela signifie qu'une MV n'a pas de relation un à un avec l'ensemble des résultats qu'elle peut avoir

Après l'analyse on peut remarquer que dans un point de vue général les mesures de "Le temps des données» et «L'état de données» puissent être deux variables de surveillance parce qu'ils sont en effet une cible valide qu'un utilisateur peut surveiller.

Dans le cas des "Niveaux d'information", il est clair que dans un environnement de BP, le détail de l'information n'est pas une variable, mais une option qui peut être choisie par l'utilisateur afin de recevoir un résultat différent pour une MV.

Si on analyse les mesures de "Le temps des données" nous pouvons voir que la dernière ligne a les types les plus représentatifs de temps puisque "Activation Time", "Completion Time" et "Duration" sont implicites dans l'ensemble composé par "Turnaround Time", "Waiting Time", "Change-over Time", "Processing Time", "Suspended Time". Chacun de ces types de temps sont une différente MV puisque chacun d'eux peut avoir une valeur pour laquelle un utilisateur peut être intéressé.

Dans le cas des mesures de "L'état de données", On peut aussi remarquer une grande diversité de valeurs dans les trois lignes, mais il est encore dans les dernières lignes où on trouve les mesures les plus représentatifs liés à l'état.

Avec cette notion, nous devons souligner que ce cas est différent aux MV que nous avons révélés auparavant, parce que l'utilisateur n'a pas intérêt de savoir les mesures de l'état.

De l'autre côté, dans un environnement de BP, les utilisateurs sont généralement intéressés de savoir l'état d'un processus ou d'une activité, alors l'état est la MV et les mesures trouvées dans le tableau composent l'espace d'état pour les valeurs possibles qui peuvent être prises par le MV de l'état.

Après l'analyse des données obtenues par un BPMS et le cycle de vie de BPM, on a l'ensemble des MV et les options de surveillance qui seront à la base de notre modèle de surveillance. Dans la section prochaine nous allons utiliser ces valeurs pour définir notre modèle.

I. 4. Le modèle de BPM

La Personnalisation de BPM est réalisé par des utilisateurs spécifiant leurs préférences ou options, sous différentes dimensions qui caractérisent le processus de surveillance. C’est pourquoi le modèle proposé définit des dimensions et des options sur la base des variables et les options de surveillance.

Une configuration personnalisable précisera les choix de l'utilisateur sur la surveillance à un moment précis dans un temps. Avec cette notion, il est clair que l'espace multidimensionnel du modèle peut être utilisé pour créer une multiple diversité de configurations.

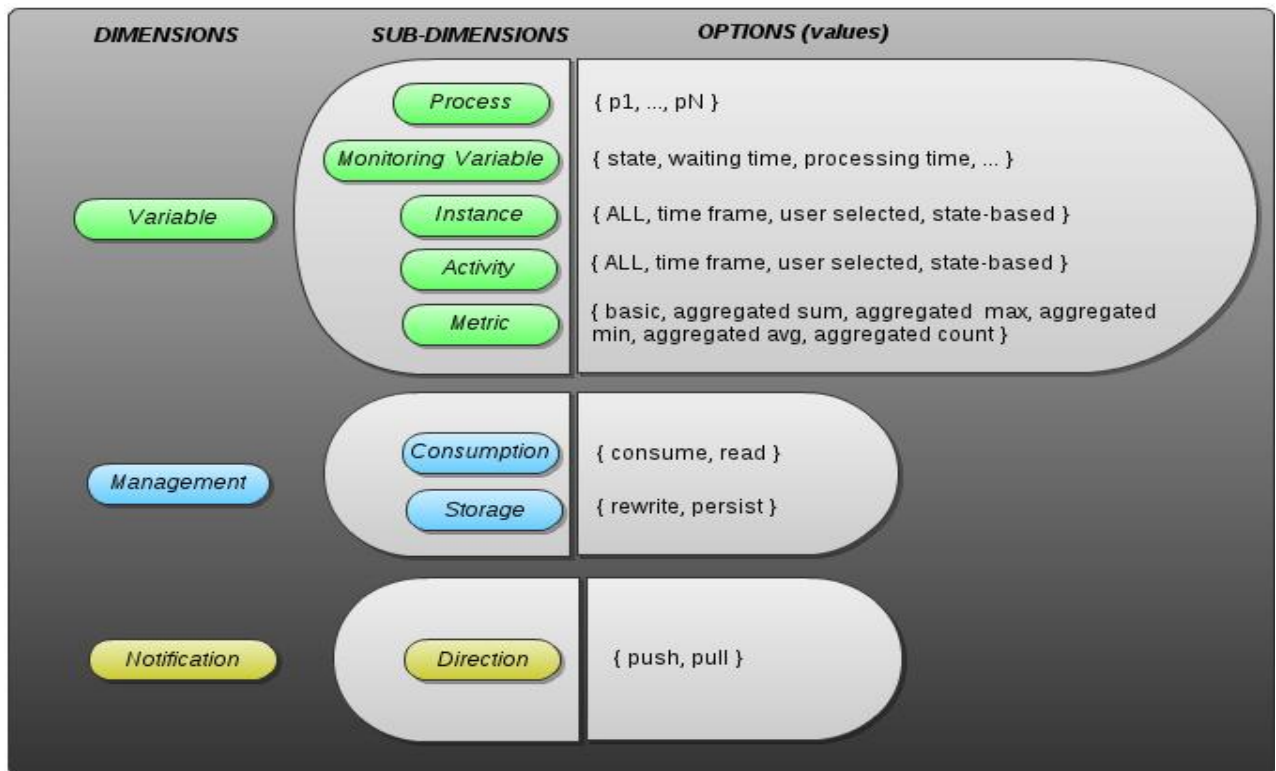


Figure I.3. Espace multidimensionnel du modèle de surveillance.

C’est, V est un tuple $\mathbf{V} = \{pro, var, inst, acte, met\}$, ou les éléments du tuple peuvent apprendre les valeurs indiquées dans la figure.1.3 est nécessaire de donner une brève clarification de ces valeurs :

Le processus p est l'identifiant du processus de K actuellement déployés dans le BPE, c’est $\mathbf{p} \in \{pk\}$, avec $k = 1, \dots, K$. Selon l'analyse effectuée dans la section précédente, la variable de contrôle est limitée à des temps des données et d'état.

Dans le cas de l'instance et de l'activité que nous considérons options qui permettent aux utilisateurs d'utiliser le temps, un état ou un spécifique critère de sélection des cas et des activités respectivement.

A propos de métrique, nous considérons soit la possibilité de voir le résultat de base de la variable sélectionnée de surveillance, ou un résultat global, qui est base dans la spécification du langage SQL opérateurs d'agrégation définition, soit $met \in \{base, \{sum, avg, max, min, count\}\}$. Notez que toutes les combinaisons d'options sont possibles.

Par exemple, il n'est pas possible d'appliquer les opérateurs d'agrégation tels que sum ou avg a des valeurs non numériques, par exemple l'activité ou de l'état de l'instance. De même, les opérateurs max et min peuvent être appliqués que sur les variables de surveillance pour laquelle une relation d'ordre total est définie.

L'élément M de spécifie les options choisies a l'égard de la dimension de gestion, tandis que l'élément N capture d'options a l'égard de la Notification dimension. Il existe deux sous dimensions de la gestion, à savoir la consommation et le stockage C S, c'est $M = (C, S)$, avec $C \in \{consommer (consume), lisez (Read)\}$ et $S \in \{réécriture (rewrite), persistant (persist)\}$. Il n'y a qu'une direction D sous dimension pour notification, avec $D \in \{pousser (push), tirer (pull)\}$.

I. 6. Conclusion

Dans ce chapitre, nous avons défini l'espace multidimensionnel des deux dimensions de surveillance et options connexes du modèle.

Dans le chapitre précédent, nous avons défini le modèle théorique de surveillance, mais pour le prouver, nous devons définir un objet tangible qui nous permet de tester la faisabilité de notre modèle dans le monde réel. Pour le tester, nous allons mettre en œuvre un outil.

Dans ce chapitre, nous allons définir l'architecture conceptuelle de l'outil.

II.1. Architecture Conceptuelle de l'Outil

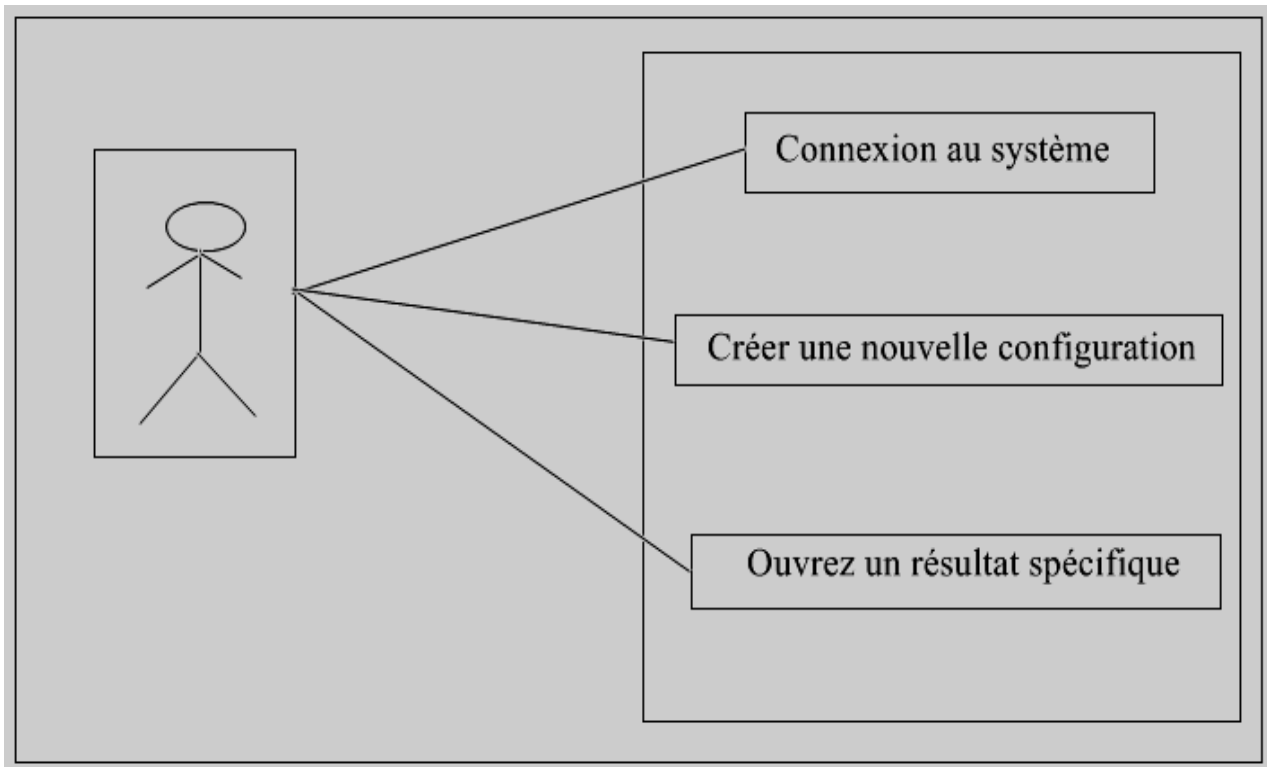


Figure II.1. Cas d'utilisation général

Avant définir l'architecture conceptuelle de l'outil, il est nécessaire de préciser que sur la base du modèle théorique présenté dans le chapitre précédent, nous pouvons déjà représenter les opérations les plus pertinentes qui devraient être pris en charge par l'outil.

Il est vraiment important de définir ces opérations de base parce qu'ils sont considérés dans la conception de notre architecture conceptuelle, qui est destinée à être utilisée comme une référence à prendre le modèle de la conception de papier à une éventuelle mise en œuvre réelle. Ces opérations sont représentées dans Figure II.1.

Compte tenu de ces opérations de base dans Figure II.1 et l'architecture général de BPM présentée dans du chapitre précédent, il est clair que le système de surveillance doit tenir compte de trois éléments :

- ✓ Client de Surveillance,
- ✓ Monitoring Infrastructure
- ✓ Moteur de processus métier (Business Process Engine).

II.2. Architecture conceptuelle

L'architecture conceptuelle de l'outil est représentée à la Figure II.2. La conception vise à isoler les composants pour chacune des dimensions de contrôle dans le modèle de contrôle.

Conceptuellement, l'interaction de l'utilisateur avec le système peut se produire au moment de la personnalisation, lors du contrôle de personnalisation sont demandé, et au moment de l'exécution, lorsque la surveillance commence [2].

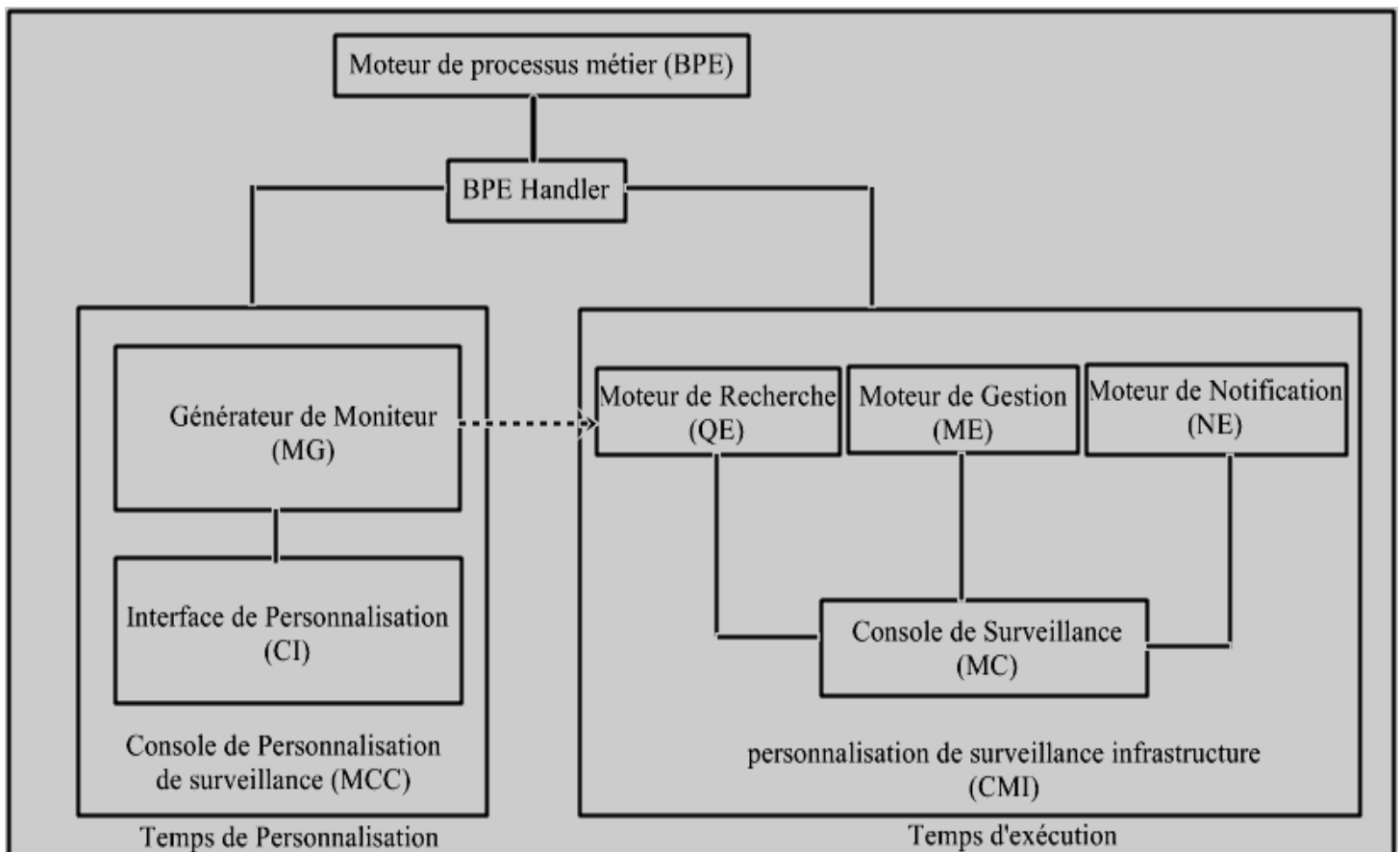


Figure II.2. Conception ET Technique

Au moment de la personnalisation, les utilisateurs accèdent à la Console de Personnalisation de surveillance (**MCC**). Celle-ci comprend l'interface de Personnalisation (**CI**), grâce à laquelle les utilisateurs précisent leurs options de personnalisation.

Les demandes de personnalisation sont reçues par le Générateur de Moniteur (**MG**), qui instancie une Personnalisation de surveillance infrastructure (**CMI**) pour chaque demande de personnalisation de surveillance.

Le CMI contient la Console de Surveillance (**MC**), grâce auquel les utilisateurs ont accès à des données de surveillance en fonction des options de surveillance spécifiées en charge de la réalisation du logique métier des options de surveillance dans notre modèle

- ✓ **Moteur de Recherche (QE)** : charge le gestionnaire BPE Handler pour capturer la variable de contrôle approprié (s) demandé dans la personnalisation.
- ✓ **Moteur de Gestion (ME)** : il réalise la logique choisie par l'utilisateur sur la gestion des données de surveillance obtenues par le BPE, ce module nécessite une base de données interne, car il peut avoir besoin pour conserver les données acquises.
- ✓ **Moteur de Notification (NE)** : il réalise la logique choisie par l'utilisateur sur la notification des résultats de la surveillance (à savoir, push et pull).

Finalement, à la fois le temps et de personnalisation des modules de temps d'exécution interagissent avec le BPE via le gestionnaire BPE. Cela agit comme la passerelle entre notre outil et la technologie des processus métier spécifique choisie pour l'exécution des processus métier surveillés.

Au moment de la personnalisation, le gestionnaire BPE permet au MCC pour obtenir des informations sur les processus actifs et les instances liées.

Au moment de l'exécution, il traduit les instructions reçues par QE en requêtes exécutables sur la base de données de journalisation du moteur de processus métier et renvoie les résultats de la requête provenant de la BPE.

Le modèle de conception de conduire la conception de l'architecture de notre outil est clairement la vue du modèle contrôleur MVC (Model-View-Controller) motif CI et MC représentent le point de vue de la personnalisation et de l'exécution, respectivement, car ils sont les interfaces à travers laquelle l'utilisateur peut interagir avec notre outil.

La représente le contrôleur dans notre architecture, reliant les interfaces à la CMI, qui met en œuvre le modèle de conduite de la surveillance du processus d'instance croix,

Par exemple, la demande de personnalisation est instancié par MG dans la logique ouvrable suivant : Compte tenu de l'élément de V, avec $V = (p1, \text{temps d'attente}, \geq 2013/12/01 \text{ ALL}, \text{avg})$. Le MG instancie la CMI ou le module MMI fait QE en mesure d'exécuter la requête suivante sur la base de données de journalisation BPE :

```
Select avg (wait_time)  
From PROCESS, INSTANCE  
Where PROCESS.ID = INSTANCE.PROC and PROCESS.ID = p1 and INSTANCE.  
Start > 2013/12/01
```

Le module MMI demande ME pour gérer l'élément M, avec $M = \langle \text{persistent}, \text{lire} \rangle$. Cela indique que l'information sera disponible dans le futur (en raison de l'option de lecture), et que l'information historique sera pris en compte, alors peu importe si l'information est tiré ou pousse dans le client de surveillance, chaque mise à jour de l'information devrait être considérer (a cause de la possibilité persister).

Dans ce cas, ME devrais avoir le logique métier pour gérer non seulement l'information, mais aussi la communication avec le client de surveillance afin d'indiquer la console de surveillance que l'information sera disponible dans le futur et que les informations historiques est considérer.

Enfin, le module MMI prie NE a manipuler l'élément N, Avec $N = \langle \text{tirez (pull)} \rangle$. Cela indique que l'option choisie était "traction informations", de sorte NE devrait indiquer le Client de Surveillance comment l'interaction sera traitée.

Une fois que NE spécifier cette option pour le client, la console de surveillance devrait fournir a l'utilisateur une méthode afin qu'il ou elle peut demander l'information.

Cet exemple montre clairement que le modèle peut dessiner facilement une configuration réelle ou il est facile de voir les parties qui composent la configuration et comment nous pouvons analyser une demande spécifique de l'utilisateur en définie actions sur les modules qui sont dans l'architecture conceptuelle.

Dans la section suivante, une analyse plus spécifique des composants de l'architecture conceptuelle est présentée.

II.3. Analyse de l'Architecture conceptuelle de l'Outil

Dans cette section, une analyse plus détaillée de l'architecture conceptuelle de l'outil sera présente. L'architecture comprend trois composantes client de surveillance, Monitoring infrastructure et le Moteur d'exécution des processus.

Le client de surveillance comprend deux modules : l'interface de Personnalisation et la Console de Monitoring. D'un point de vue technique, ces deux composants peuvent être définis comme une interface qui permet à l'utilisateur de régler une demande et de voir les résultats de la demande.

L'infrastructure de surveillance est composée de plusieurs modules. L'un des deux est le générateur moniteur (MG), qui établit une communication directe avec le MC. MG identifie le type de demande et instancie la Personnalisation de surveillance infrastructure (CMI).

La principale composante de la CMI est le module de surveillance de modèle d'intégration (MMI), qui coordonne et gère l'exécution des trois autres composantes : moteur de recherche (QE), Moteur de gestion (ME) et moteur de notification (NE).

Lorsque MMI finit l'exécution d'une demande, il gère le résultat et le transfère à la MG qui a la logique pour alimenter le MC et afficher les résultats à l'utilisateur final.

Selon le modèle, Moteur de gestion (ME) est un composant qui a la logique de fournir les options de la sous-dimension consommation et Stockage. Ces options ne nécessitent pas une interaction avec le BPE.

Moteur de notification (NE) :

Est un autre module fournir les options de la sous-dimension "Direction". Il est également vrai que ces options ne nécessitent pas d'interaction avec le BPE.

Au lieu de cela, les options de "(pull)" et "(push)" déterminent la façon dont l'information sera flux entre l'infrastructure de surveillance et le client de surveillance.

Moteur de recherche (QE) :

Est un module important du modèle, car ce composant s'occupe d'établir la connexion entre le CMI et le BPE. Comme il a été expliqué dans le chapitre précédent, la connexion est conçue pour être à un niveau d'interaction de base de données.

Jusqu'à présent, nous avons analysé d'un point de vue technique toutes les composantes de notre architecture conceptuelle et il est possible de dire que le modèle semble faisable.

Dans la section suivante une architecture technique proposée sera définie et au final un outil concret sera mis en œuvre sous la spécification de la finale architecture technique. Cet outil nous aidera à conclure si le modèle est vraiment faisable ou non.

II.4. Architecture Technique

En 1980, le langage de programmation Small talk, un langage orienté objet, montre une fonctionnalité qui est devenue vraiment populaire jusqu'à nos jours puisque les entreprises importantes, telles que Sun Microsystems a décidé de l'utiliser pour ses boîtes à outils et des applications d'entreprise.

Cette fonctionnalité est le modèle Model / View / Controller (MVC). Cette tendance a été vraiment utile dans l'architecture de plusieurs systèmes d'information. MVC également connu comme Présentation / Abstraction / Contrôle modèle de conception a pour principe de base pour séparer les interfaces utilisateur à partir des données affichées par l'interface utilisateur.

Le flux du motif est représenté dans la Figure II.3.

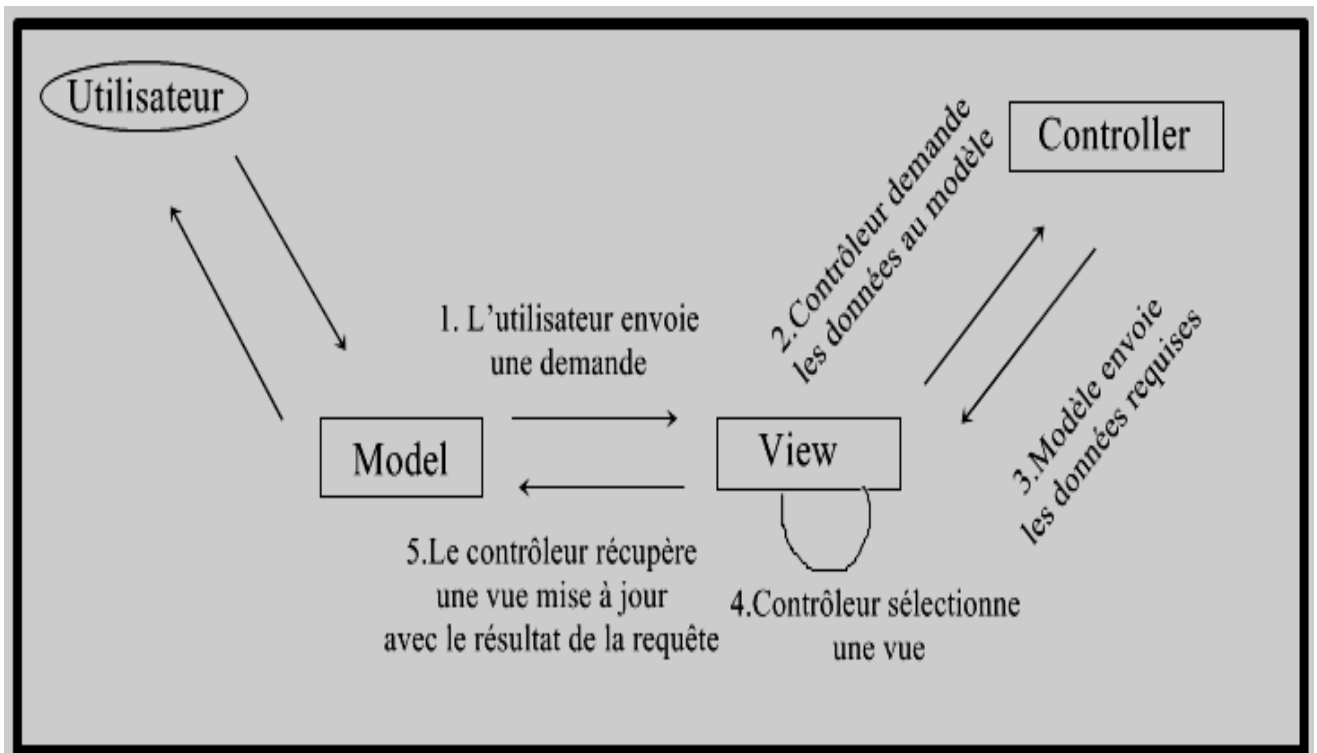


Figure II.3. Modèle MVC Model / View / Controller MVC

Il ya trois éléments majeurs dans l'architecture MVC :

- **Model** élément qui encapsule le « business logic » pour gérer le comportement et les données de l'application.
- **View** l'élément qui contient la représentation des données dans le modèle. Cet élément représente la présentation de la demande et ne dépend pas de la logique d'application.
- **Controller** : élément qui communique et coordonne les activités entre le model et la View. Cet élément intercepte essentiellement les demandes de la vue et les transmet au modèle d'exécuter les mesures appropriées pour répondre à la demande.

L'objectif principal de l'utilisation d'une architecture MVC est la séparation entre les deux données d'application et le business logic de la couche de présentation à l'utilisateur. Les vrais avantages de l'utilisation du modèle de conception MVC sont:

- ✓ Réutilisation du business logic.
- ✓ Souplesse d'utiliser plusieurs et différents types d'interfaces d'utilisateur.

- ✓ Spécialisation dans le développement (En termes de développement de logiciels).
- ✓ Développement parallèle (En termes de développement de logiciels).

Maintenant, avec la définition claire du modèle MVC, il est facile de voir que notre architecture conceptuelle est composée par des composants qui peuvent être mappées à une architecture MVC.

Cette intégration est représenté e à la Figure II.4 et il est basé sur les arguments suivants :

Monitoring Client MC–View :

Le client de surveillance est l'interface avec l'utilisateur final. MI permet à l'utilisateur soit de choisir une configuration ou pour voir les résultats de configurations déjà définies dans le module de la console de surveillance.

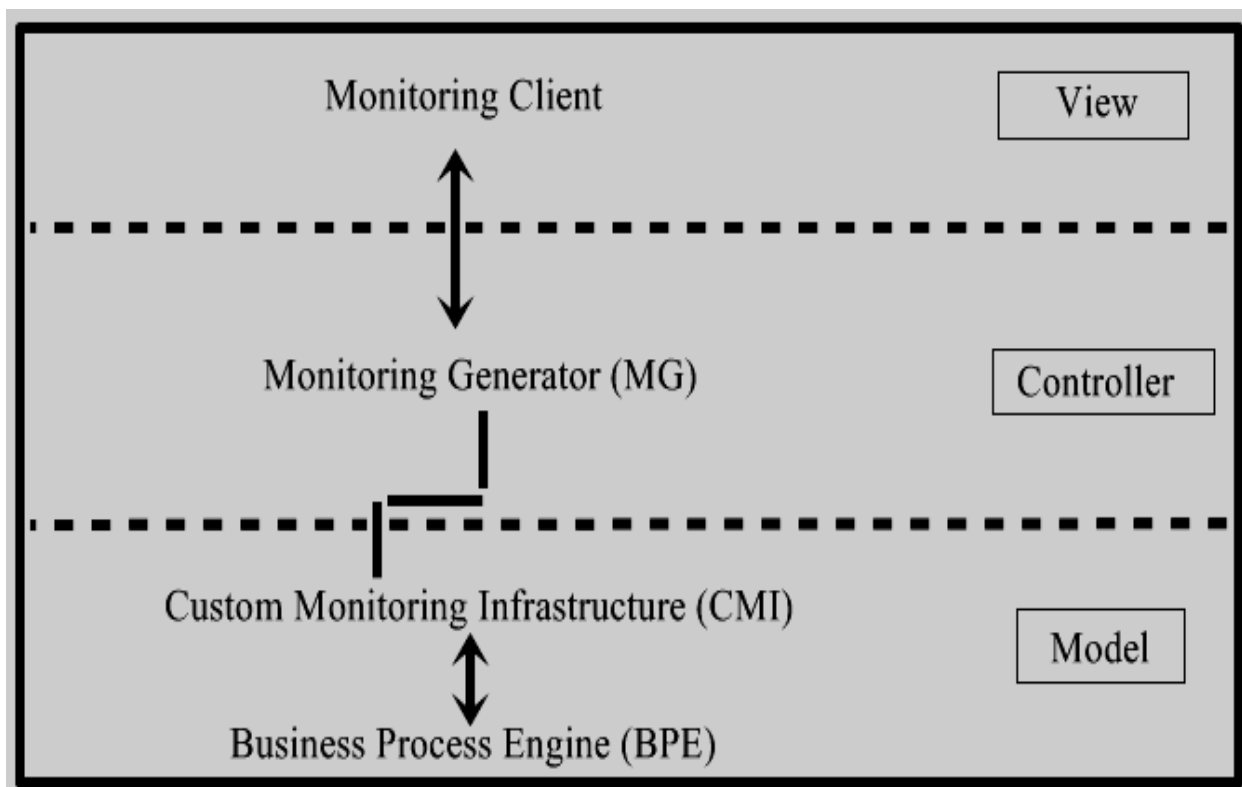


Figure II.4. MVC appliquée à l'architecture conceptuelle

Personnalisation de surveillance infrastructure CMI–Model :

Personnalisation de surveillance infrastructure non seulement examiner le contenu et d'information des options de récupération, mais aussi extraire et gérer les données de la base BPE.

Générateur de Moniteur MG – Controller :

Le Générateur de Moniteur est le point de connexion entre le MC et le CMI. Dans cette section, nous avons analysé l'architecture conceptuelle et cartographié à l'architecture logicielle réelle et prouvée comme la MVC.

Dans la section suivante, nous allons utiliser l'un des Framework basés sur le modèle MVC pour créer l'outil qui sera utilisé pour tester le modèle dans un environnement d'exécution de processus métier réel.

II.5. Technique Réalisation

L'outil a été mis en œuvre comme une application web, Nous avons expérimenté avec le moteur Apache ODE BPEL, qui est disponible gratuitement et open source, cet est un gestionnaire BPE spécifique a été développé pour Apache ODE.

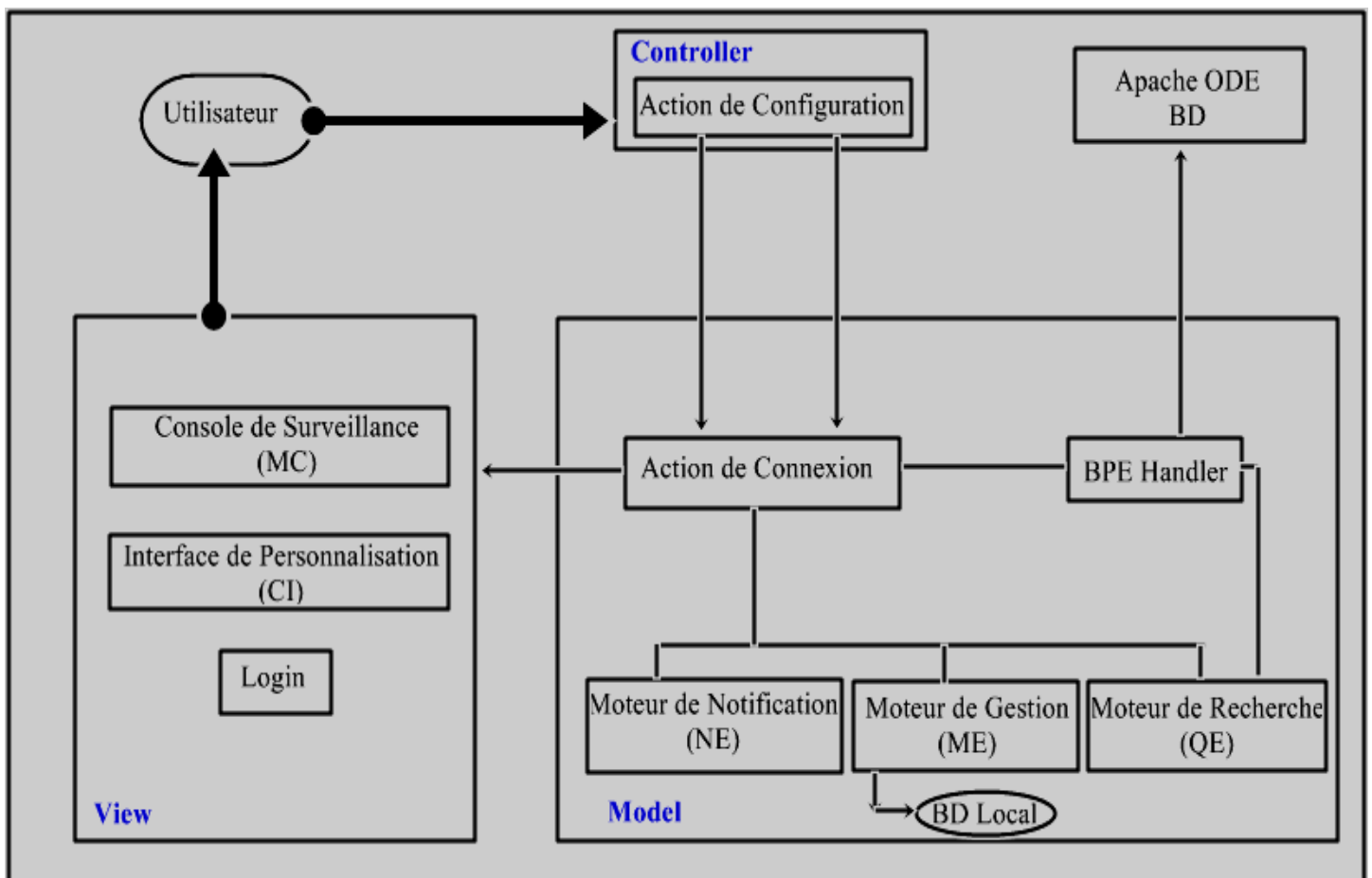


Figure II.5. L'architecture mis en place de l'outil

La Figure II.5 montre l'architecture technique de notre outil, le composant MG dans l'architecture conceptuelle est mis en œuvre par le composant de contrôleur.

Le contrôleur met en œuvre le logique métier de l'infrastructure de surveillance personnalisé. Spécifiquement, le BPE interroge le gestionnaire des requêtes sur la base de données Apache ODE, alors que la composante du moteur de gestion (Management Engine) contrôle une base de données locale où les données acquises d'Apache ODE peuvent être stockés, l'utilisateur demande des données de surveillance d'être persisté.

La vue de notre outil comprend un ensemble de pages Web dynamiques, qui sont configurés par les données demandées par l'action mise en œuvre dans le contrôleur.

L'outil comprend une page pour la connexion de l'utilisateur et les pages implémentant l'interface de personnalisation et la console de surveillance.

II.6. Conclusion

Dans ce chapitre nous pouvons encore résumons que le modèle et son espace multidimensionnel sont la partie principale des réponses, tout simplement parce qu'ils montrent clairement non seulement les informations clients qui devrait atteindre d'un BPMS, mais aussi quels sont les moyens possibles, ou des options, que les utilisateurs peuvent obtenir.

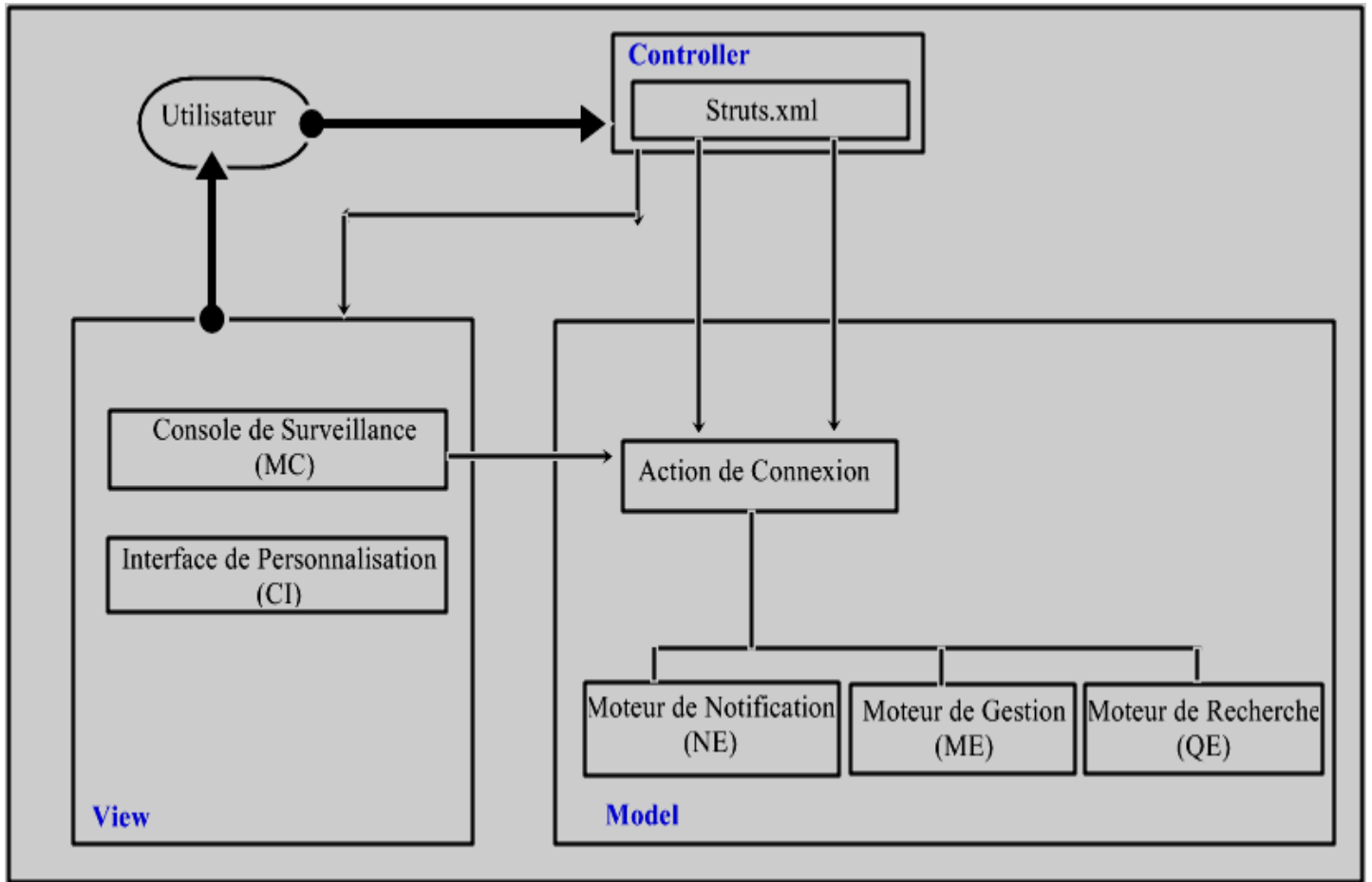


Figure III.2. Cartographie de l'architecture MVC du modèle dans la MVC.

- **Monitoring Client** : deux composantes principales : L'interface de Personnalisation (CI) et Console de Surveillance(MC).
- **Générateur de Moniteur** : peut être affecté à la composante de contrôleur de l'architecture MVC, donc dans la mise en œuvre finale, il n'y aura pas un composant avec le nom "Monitoring Générateur". Dans ce cas, le composant MG sera remplacé par le "struts.xml", le noyau configuration fichier pour Framework.
- **Personnalisation de surveillance infrastructure (CMI)** : sera mis en œuvre avec un ensemble de classes d'action et autres classes avec le business logic de la Moteur de Gestion, Moteur de Notification et Moteur de Recherche.

Dans la section suivante, nous allons fournir une analyse approfondie de l'outil qui a été mis en œuvre afin de prouver notre modèle dans un environnement réel.

III.2. Spécifications de la mise en œuvre réelle

Cette section fournit une analyse approfondie de l'outil qui a été mis en œuvre afin de prouver le modèle proposé de surveillance de ce projet dans un environnement réel.

BPEL :

Dans le cas spécifique de la mise en œuvre de l'outil qui se révélera notre modèle, nous avons décidé d'utiliser un moteur BPEL. Avant d'expliquer plus de détails sur l'outil mis en œuvre, il est nécessaire de donner une brève explication sur BPEL.

BPEL (Business Process Execution Language) est une norme d'un grand potentiel pour les entreprises à atteindre leurs objectifs commerciaux et améliorer leurs processus en intégrant une variété d'applications, indépendamment des plates-formes et technologies associés à chaque activité.

Originellement écrit par des entreprises telles que BEA Systems, IBM et Microsoft, avec le soutien d'OASIS (Organisation pour la promotion de la Structured Information Standards), BPEL atteint un fort soutien de l'industrie de la technologie, en particulier dans les grandes entreprises, telles qu'Oracle et Sun, ainsi Novell, Adobe et SAP, entre autres.

En outre, il permet aux entreprises d'atteindre un haut dynamisme technologique dans son architecture, adapté rapidement au changement, que ce soit en interne ou en externe. De cette façon, ils peuvent plus facilement réorganiser la communication entre les applications, réduisant de manière significative la complexité des processus.

La popularité de BPEL est basée sur le fait qu'il est un langage XML qui définit comment un processus métier peut être implémenté en utilisant les services Web. Quand un processus métier est mis en œuvre par les services Web à travers BPEL cela signifie qu'il y aura une interface unique pour soutenir messages XML indépendamment de la plateforme associée, éliminant ainsi la nécessité d'utiliser plusieurs protocoles et des formats et différentes interfaces.

Même lorsqu'ils ne sont pas toutes les activités sont actuellement mis en œuvre en tant que services Web dans les organisations, les effets sont tangibles en interne, car ils aident à simplifier et rendre l'interaction plus rapide et l'exécution d'un processus métier.

Avec cette notion, BPEL est suffisante pour définir un flux de processus, la situation pourquoi il est considéré comme une technologie clé pour le présent et l'avenir proche de l'industrie.

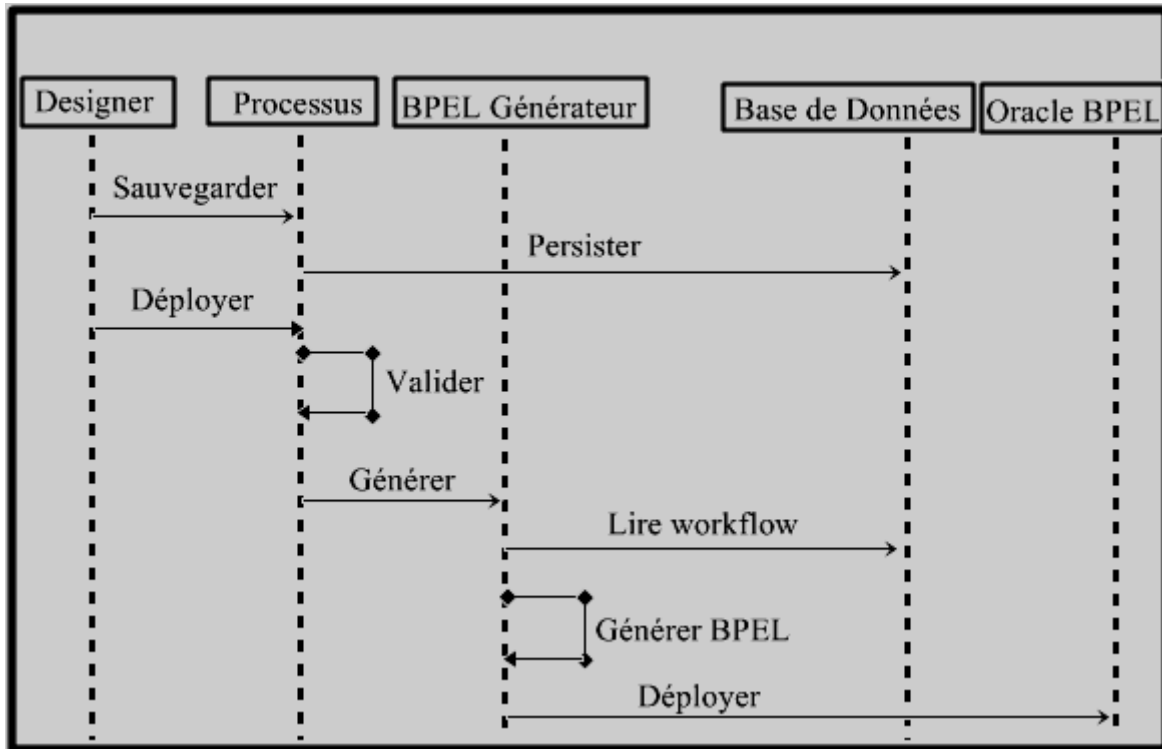


Figure III.3. Construire avec processus BPEL.

Les avantages d'intégration qui fournit BPEL, il donne à l'utilisateur la possibilité de construire sur processus. Selon la référence [4], l'écoulement pour créer un processus sur l'ordonnée peut être décrit sur la Figure III.3 et comprend les étapes suivantes :

1. Utilisation d'un concepteur de modéliser graphiquement les processus métier.
2. La définition du processus est stockée dans la base de données par le créateur
3. Le BPEL Générateur non seulement valide la définition de processus, mais aussi génère à la fois la Fichier XML BPEL de la représentation de base de données et les fichiers associés utilisés pour le déploiement.
4. Le processus BPEL est dynamiquement déployé sur le processus BPEL Server.

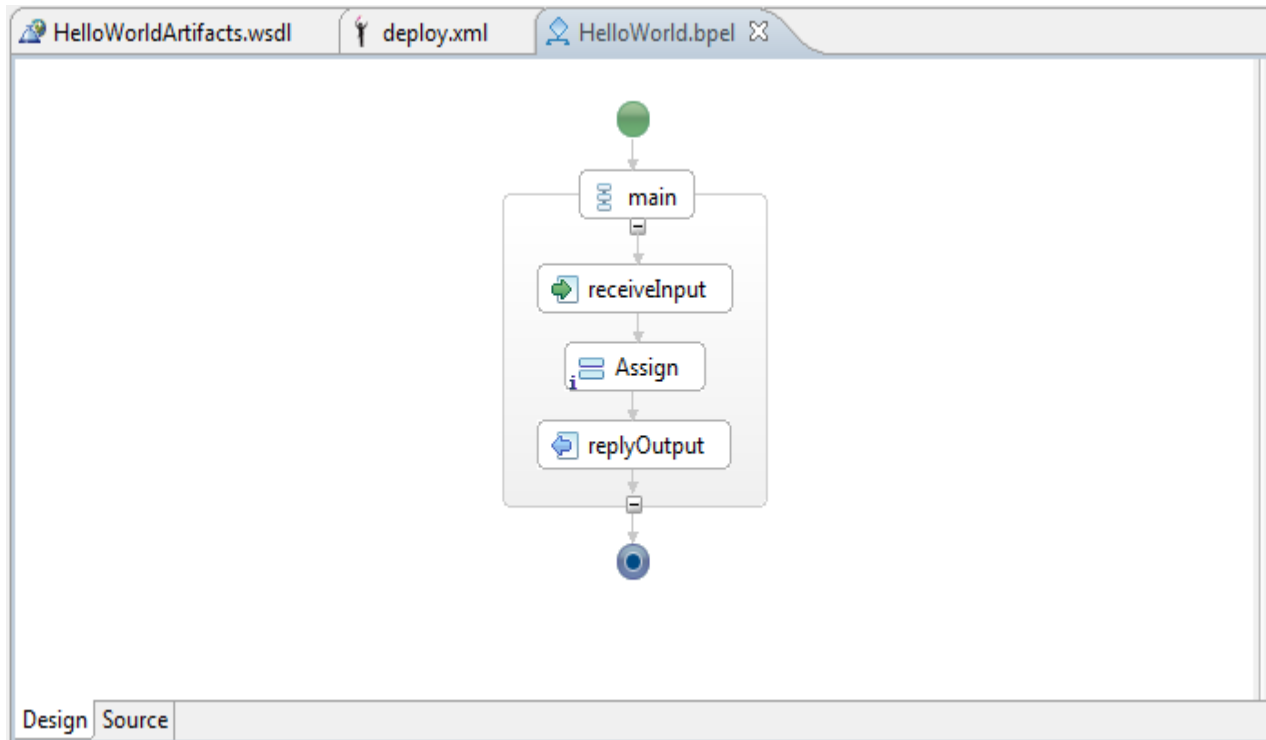


Figure III.4. Construire un processus BPEL avec Eclipse éditeur Designer.

Maintenant que nous avons expliqué pourquoi BPEL a été choisi, nous pouvons commencer à expliquer comment était la mise en œuvre réelle. Comme il est indiqué dans la liste des étapes pour construire un processus, nous pouvons utiliser un concepteur et nous avons choisi BPEL Designer éditeur, un projet open source dans le cadre du Projet de technologie Eclipse.

Cet Editeur Designer offre un soutien pour l'intégration BPEL4WS expérience de conception avec la plate-forme Eclipse. La Figure III.4 montre que l'on peut facilement créer un processus métier avec l'éditeur.

Jusqu'à présent, il est clair que nous pouvons créer des processus métier BPEL avec l'éditeur de designer, mais ces processus ont besoin d'un environnement où ils peuvent être exécutés. Pour les processus métier BPEL, cet environnement est Business Process Execution Runtime (BPER).

Un BPER a deux principales tâches à soutenir. Le premier est d'exécuter du processus, le stockage de l'état des processus. Ce dernier est chargé de la communication entre les processus et services. La BPER choisi pour ce projet est le moteur d'orchestration d'Apache.

Selon [5] "Apache ODE est un open-source BPER et utilise moteur Apaches Web Service Axis2 Web pour réaliser la communication de Web Service".

III.3. Analyse détaillée de la mise en œuvre

Maintenant nous avons une idée claire du moteur que nous utilisons, nous pouvons expliquer avec plus de détails la façon dont l'outil a été vraiment mis en œuvre. Figure III.8 représenté l'architecture de l'outil mis en place, que nous appelons "BP Monitor".

Il est clair que la Figure III.8 est une vue plus spécifique de la cartographie de l'architecture présentée dans la Figure III.2 de la section précédente.

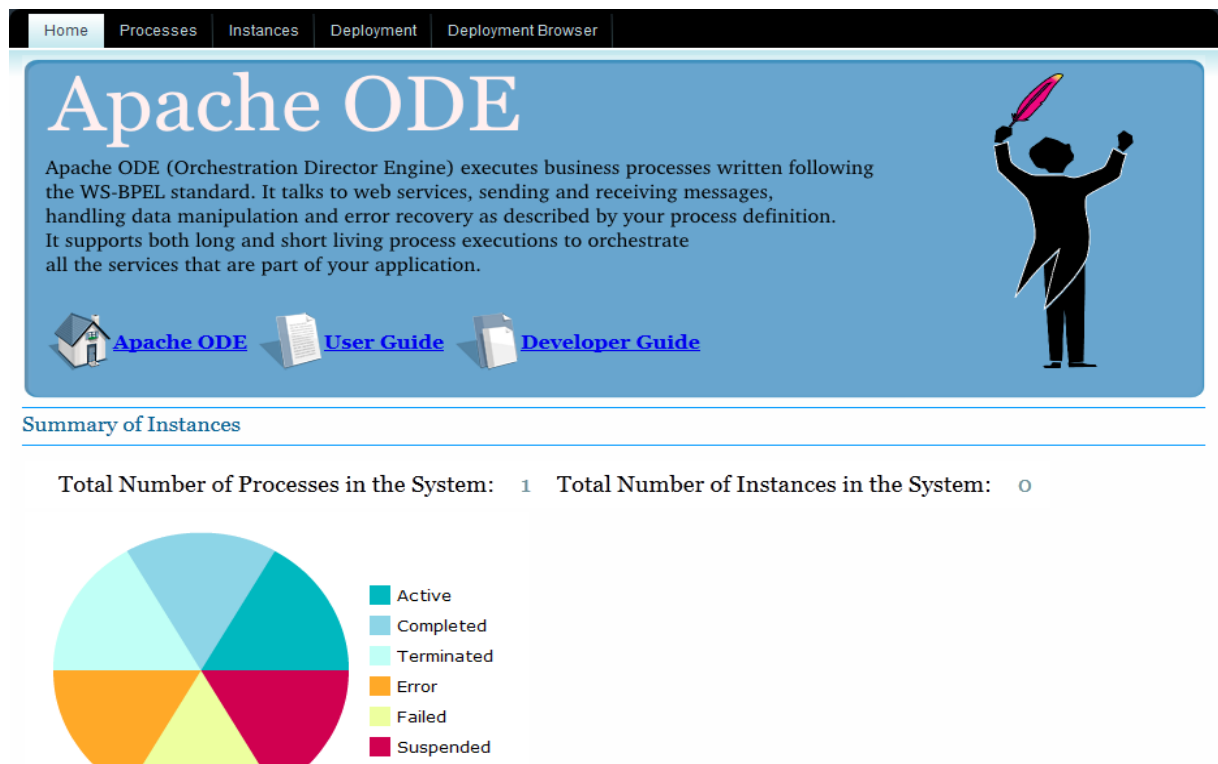


Figure III.5. Console Web Apache ODE (page d'accueil).

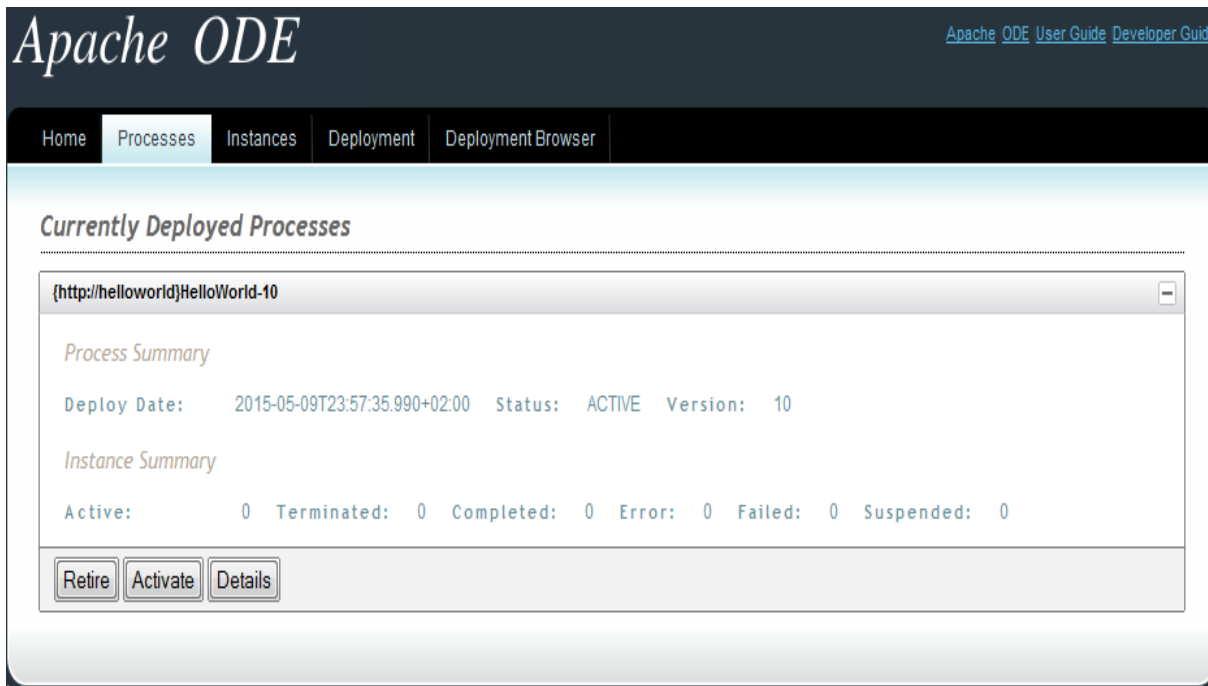


Figure III.6. Console Web Apache ODE (page de processus).

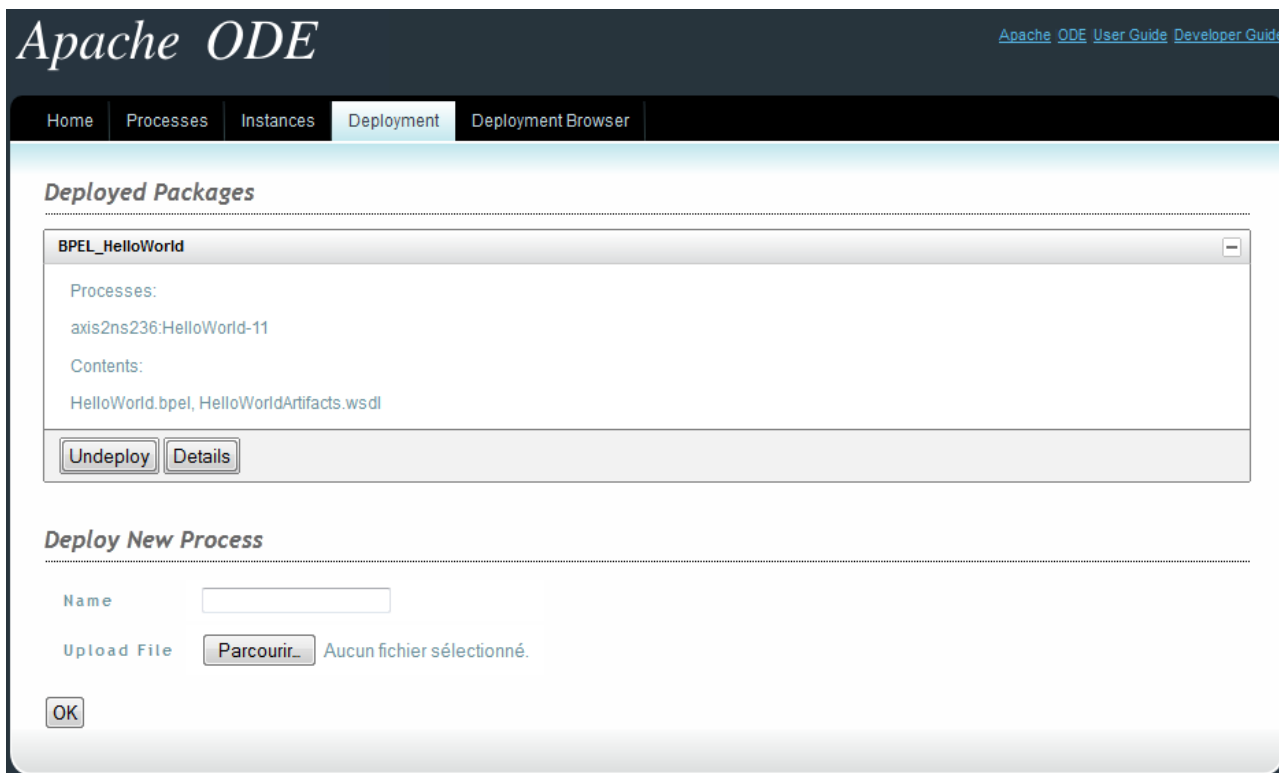


Figure III.7. Console Web Apache ODE (Page des Cas).

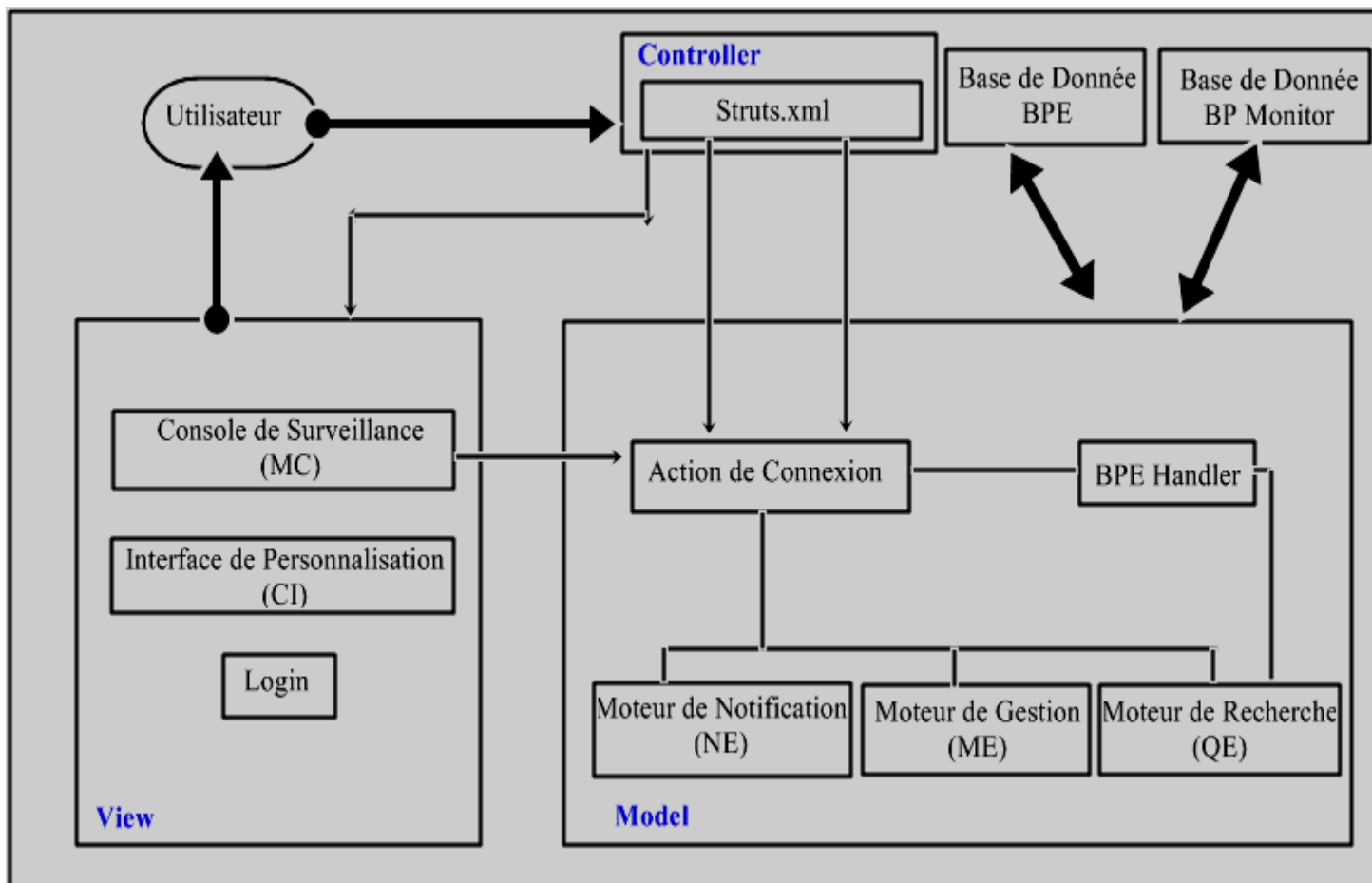


Figure III.8. L'architecture détaillée de mis en œuvre l'outil (BP Monitor).

Deux éléments que nous trouvons dans la Figure III.8 sont des bases de données. La première base de données est associée au moteur BPEL et celui-ci à la base de données interne qui supporte l'outil mis en œuvre.

Lorsque nous passons en revue la documentation d'Apache ODE nous constatons qu'ODE assure la fiabilité de l'exécution des processus en utilisant une base de données relationnelle pour fournir un stockage persistant. Après une analyse et en tenant compte des variables de surveillance qui sont utilisés par le modèle que nous avons décidé que nous allons utiliser les tableaux représentés dans la Figure III.9.

Figure III.10 représenté le diagramme ER de la base de données interne utilisée par BP Monitor.

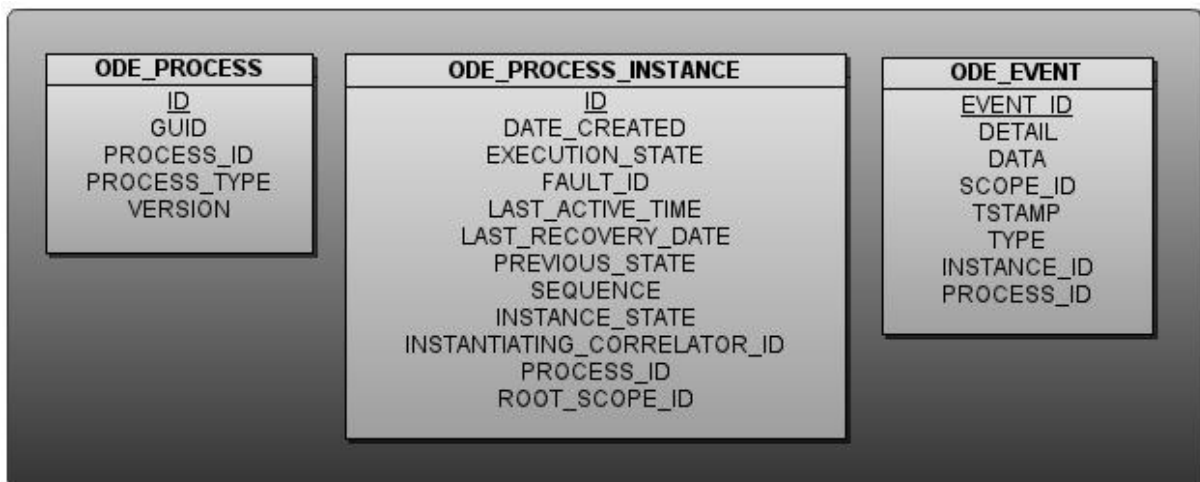


Figure III.9. Modèle Apache ODE ER (Tables d'intérêt).

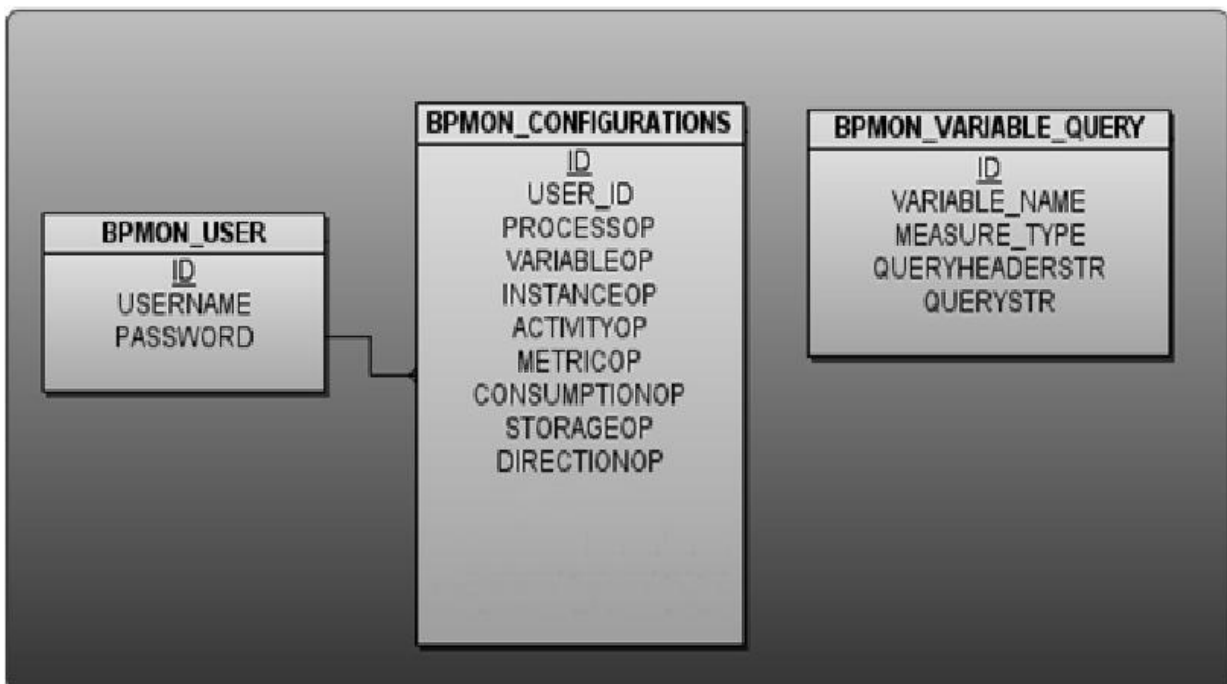


Figure III.10. Modèle ER de BP Monitor.

Lorsque nous analysons le composant de vue de l'architecture détaillée, nous voyons trois éléments principaux : Interface de personnalisation, Monitoring Console, et Login.

L'interface de connexion est utilisée pour identifier l'utilisateur qui accède à l'outil. Cette interface est vraiment importante puisque le but de l'outil est de fournir des informations personnalisables monitoring non seulement à un seul utilisateur, mais à un ensemble de différents utilisateurs, montrant un autre avantage que notre modèle a sur les consoles de surveillance standard.

L'interface de Personnalisation est représentée dans la Figure III.11. Cette interface présente les configurations que l'utilisateur a déjà créés et lui permet ou elle pour créer de nouvelles spécifications de personnalisation (Nous allons utiliser le terme "configuration" de faire une référence à une spécification de personnalisation spécifique).

Si nous analysons la Figure III.11, nous allons voir qu'il a clairement une interface où l'utilisateur peut trouver l'espace multidimensionnel de deux dimensions et les options de surveillance que nous avons définie dans le chapitre précédent.

Cette interface non seulement correspond directement au modèle défini dans le chapitre précédent, mais permet également à l'utilisateur d'avoir la flexibilité suivante quand il crée une configuration :

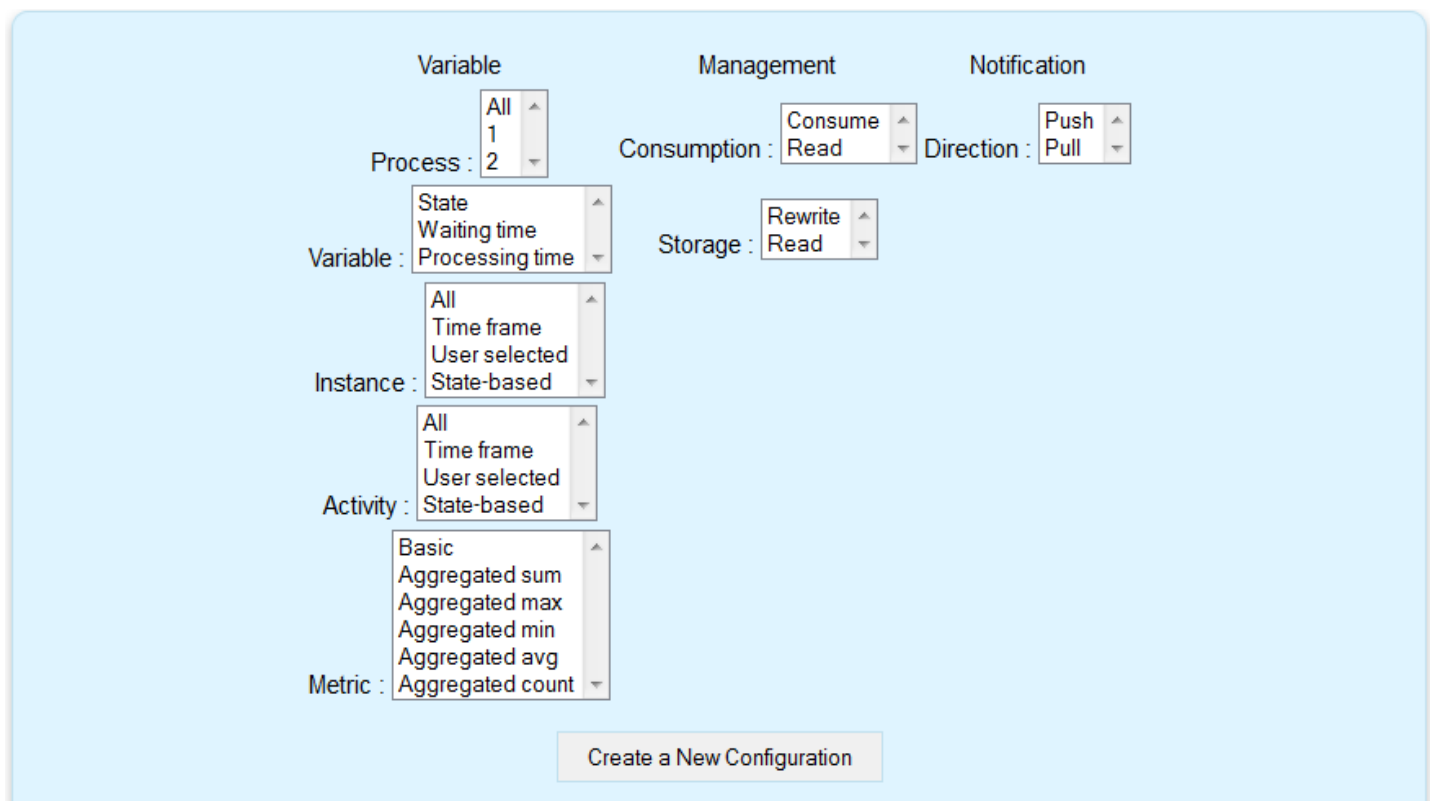


Figure III.11. BP Monitor - Personnalisation de l'interface

Processus L'utilisateur peut sélectionner l'un, un ensemble ou tous les processus.

Variable utilisateur peut sélectionner une ou plusieurs variables de contrôle dans le même temps.

Instance utilisateur peut sélectionner un critère pour choisir des cas spécifiques (tous les cas, les instances d'utilisateur spécifié, ou des instances en fonction du temps)

Activité L'utilisateur peut sélectionner un critère pour choisir des activités spécifiques (toutes les activités, les activités spécifiées par l'utilisateur, ou d'activités en fonction du temps)

Matrice utilisateur peut sélectionner une métrique de base ou une ou des mesures plus agrégées dans le même temps.

Consommation L'utilisateur peut sélectionner soit «consommer» ou «lire», mais pas les deux.

Storage L'utilisateur peut sélectionner soit "réécrire" ou "persister", mais pas les deux.

Direction L'utilisateur peut sélectionner soit "push" ou "pull", mais pas les deux.

Il est nécessaire de préciser que chaque fois que l'utilisateur est connecté BP Monitor, il sera toujours voir l'interface de Personnalisation, puis l'utilisateur peut décider de créer de nouvelles configurations ou tout simplement passer à Console de Surveillance et de voir les résultats pour les actifs configurations.

Le modèle a plusieurs composants tels que Moteur de Notification, Moteur de Gestion et Moteur de Recherche, Toutefois, dans la mise en œuvre définitive du processus métier a été noyée dans le corps des actions Struts2, qui sont essentiellement un ensemble de classes Java.

En outre, nous trouvons un BD Helper composante qui est essentiellement chargée d'établir la communication entre l'outil de monitoring et les bases de données qui ont été décrites précédemment.

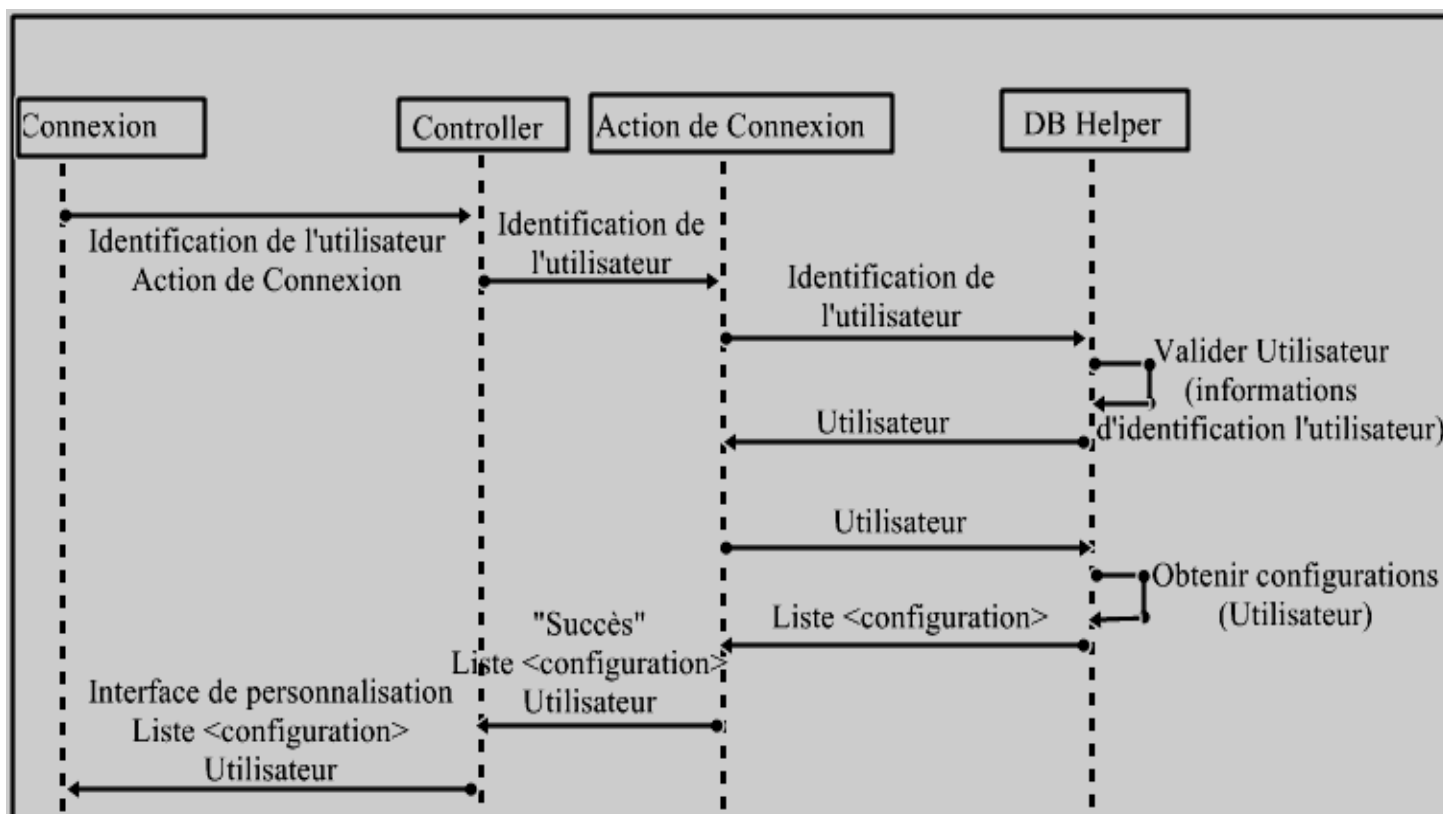


Figure III.12. BP Monitor - Interface de Connexion - Connexion

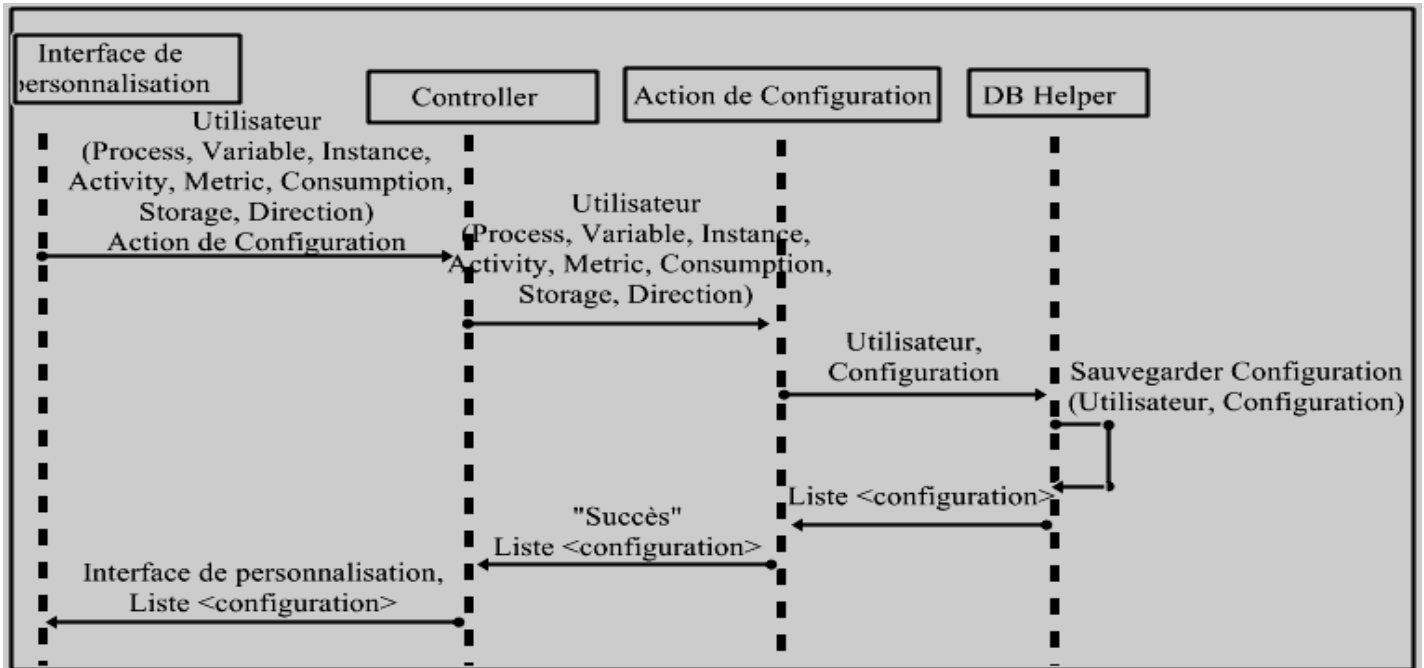


Figure III.13. BP Monitor - l'interface de Personnalisation - Créer Configuration

Le flow à l'intérieur de mis en œuvre de l'outil (BP Monitor) est décrit dans les points suivants :

Connexion au système La première étape pour utiliser l'outil de surveillance est de fournir les informations d'identification de l'utilisateur afin que BP Monitor valide et permet l'accès.

Si l'utilisateur est un utilisateur valide pour le système l'action Connexion demandera à l'instance BD Helper la liste de toutes les configurations créées par l'utilisateur dans le passé. Il est important de dire que l'action de Connexion est une partie des moteurs (QE, ME, et NE) parce que cette action vérifie la liste des configurations et analyse chacun d'eux afin de trouver l'V, M et N dimensions.

Une fois que chaque configuration est analysée, l'action Connexion va filtrer les configurations en enlevant ceux qui ont une valeur différente de «lire» dans la sous-dimension Consommation de la dimension M.

Lorsque la liste des configurations est filtrée l'action Connexion redirigera l'utilisateur vers l'interface de personnalisation, représenté dans la Figure III.11, qui montrera les configurations filtrés à l'utilisateur. Ce qui est représenté dans la Figure III.12.

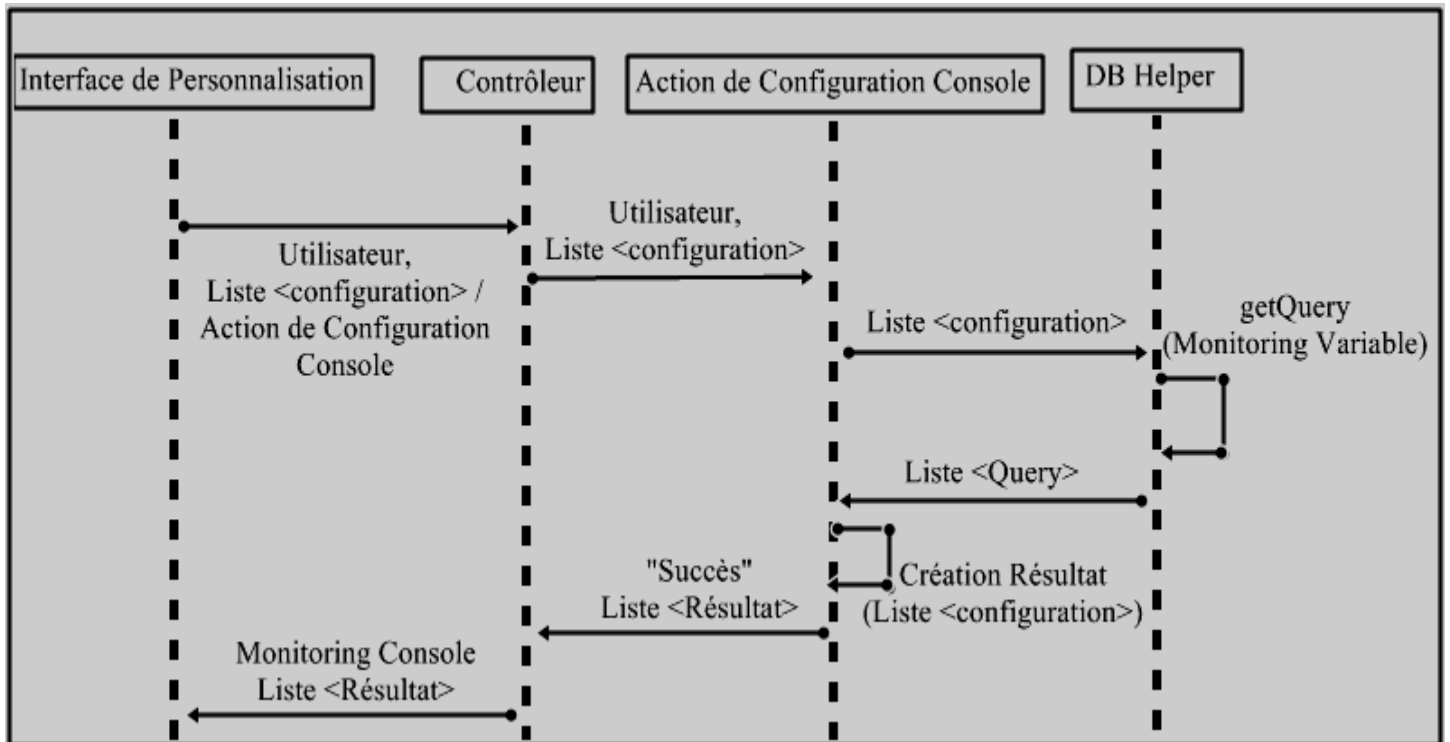


Figure III.14 BP Monitor - l'interface de Personnalisation - Obtenez des résultats

Créer une nouvelle configuration Une fois que l'utilisateur a ouvert une session dans le système, il va voir l'interface de Personnalisation représenté dans la Figure III.11, où il peut non seulement voir les configurations qui ont déjà été créés, mais aussi créer de nouvelles configurations.

Quand l'utilisateur sélectionne des valeurs différentes dans la l'interface de personnalisation et quand il décide de créer une nouvelle configuration de l'action Configurer l'action sera instancié. Configurer Action utilise les valeurs sélectionnées pour identifier les dimensions :

V = (Process value; Variable value; Instance value; Activity value; Metric value)

M = (Consumption value; Storage value)

N = (Direction value)

Lorsque les dimensions sont identifiées l'action Configurer peut créer une nouvelle instance de configuration de type, objet qui sera envoyé à BP Helper afin de sauvegarder cette nouvelle configuration dans la base de données interne de l'outil.

Une fois la nouvelle configuration est stockée, BD Helper récupère la liste mise à jour des configurations qui seront mis à jour dans l'interface de personnalisation, qui mettra à jour l'utilisateur sur l'état de la transaction.

Après cette action, l'utilisateur est libre de répéter le processus en créant plusieurs configurations ou saut dans la console de surveillance pour voir les résultats pour les configurations actives actuelles. Le flux pour créer une nouvelle configuration est représenté dans la Figure III.13.

Sautez dans la Console de Surveillance : Une fois que l'utilisateur a ouvert une session dans le système, il verra l'interface de personnalisation représenté, où il peut sauter dans Monitoring Console, où les résultats pour chaque configuration qui est représentée dans la Figure III.14.

Lorsque l'utilisateur décide de sauter dans la console, la console action configure de boucle dans toutes les configurations, identifier les trois principales composantes V, M, N et dans chacun d'eux.

Comme il a été expliqué dans le chapitre précédent, la composante V doit être mappé dans une requête SQL, et ceci est la raison pour laquelle dans la Figure III.14 instance Configure Console Action demande une liste de requêtes SQL pour l'instance BD Helper, qui est la classe qui communique avec la base de données interne afin d'obtenir les requêtes SQL qui correspondent à une variable d'un monitoring spécifique.

Lorsque Configure Console Action a spécifique la requête pour une variable de surveillance, il peut appliquer les paramètres sélectionnés tels que les processus, les instances, et / ou d'activités afin d'avoir une requête SQL qui considère vraiment toutes les valeurs qui sont en V.

La composante de M qui devrait être analysé par le Moteur de Gestion devrait également être prise en charge par l'instance Action de Configure Console, dans la mise en œuvre réelle nous avons fait des hypothèses et l'un d'eux est d'appliquer une valeur de "réécriture" du stockage dimension dans tous les cas, ce qui signifie que nous ne considérerons pas l'information historique, de sorte que l'outil affiche les informations de la demande la plus récente.

La composante N, qui devrait être analysée par Moteur de Notification, est également prise en charge par Configure Console Action instance qui identifie et crée des interfaces Web, avec le code approprié pour soutenir la traction de l'information des options ou des informations poussée en fonction de la valeur de N.

Ouvrez un résultat spécifique Une fois l'utilisateur voir la Monitoring Console, il peut voir la liste des configurations et chaque configuration a un ensemble de liens, chaque lien est une référence à un résultat spécifique. Cette section a donné une explication détaillée des composants finaux et le réel flux mis en œuvre de l'outil.

I.4. Conclusion

Dans ce chapitre, nous avons donné une explication profonde de la façon dont le modèle théorique, présenté dans le chapitre précédent, peut être prise à partir de la conception de papier à une mise en œuvre réelle.

Ce processus a impliqué non seulement la définition et la cartographie de l'architecture conceptuelle d'une architecture technique, mais aussi la cartographie de l'architecture technique proposée dans un cadre de développement logiciel réel, et la mise en œuvre du cadre.

CONCLUSION

Notre travail avait pour but d'analyser la relation entre "la gestion des processus métiers", en particulier d'un point de vue gestion, et spécifiquement de "*Business Process Monitoring*" et la "*Personnalisation*".

Nous pouvons constater que même s'il est possible de trouver non seulement une grande diversité de la littérature de recherche, mais aussi une large gamme d'applications réelle pour chacun d'eux séparément, la vérité est que leur relation n'a pas été étudiée par la littérature.

D'ailleurs, il est possible de voir que la technologie actuelle offre des options limitées pour la personnalisation du suivi des processus métiers, et que même lorsque la technologie prévoit déjà des standards comme BAM, ils offrent une vision limitée de l'information. En réalité, aucune de ces options ne représente un modèle personnalisable avec la flexibilité nécessaire pour répondre aux besoins de tous les clients concernés dans le cadre de la surveillance des processus métiers.

Ces arguments sont la principale raison que notre définition du problème est de trouver la réponse à la question suivante : "Est-il possible de définir un modèle de surveillance personnalisable pour les processus métier ?" La réponse à cette question peut être trouvée dans le chapitre où l'étude Littérature et le modèle théorique sont présentés.

REFERENCES

- [1]: **Marco Comuzzi, Samuil Angelov, and Jochem Vonk.** *Patterns to enable mass-customized business process monitoring.* In *24 th International Conference on Advanced Information Systems Engineering (CAiSE'12), 2012.* iii, 8, 15, 16
- [2]: **F. Gottschalk, W. van der Aalst, M. Jansen-Vullers, and M. La Rosa,** “*Configurable workflow models,*” *IJCIS, vol. 17, pp. 177–221, 2008.*
- [3]: **Ian Roughley.** *Starting Struts2.* *Lulu.com, 200.* 29
- [4]: **Arun Poduval, Doug Todd, Harish Gaur, Jeremy Bolie, Jerry Thomas, Kevin Geminiuc, Lawrence Pravin, Markus Zirn, Matjaz B. Juric, Michael Cardella, Praveen Ramachandran, Sean Carey, Stany Blanvalet, The Hoa Nguyen, and Yves Coene.** *BPEL Cookbook: Best Practices for SOA-based integration and composite applications development.* *Packt Publishing, 2006.* 31
- [5]: **Nils Glombitza, Dennis Pesterer, and Stefan Fischer.** *Integrating wireless sensor networks into web service-based business processes.* In *Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks,* pages 25-30. *ACM, 2009.* 32
- [6]: **M. zur Muehlen and R. Shapiro,** *Business process analytics,* in *Handbook on Business Process Management 2, 2010,* pp. 137–157.
- [7]: **L. Baresi and S. Guinea,** *Self-supervising BPEL processes,* *IEEE Transactions on Software Engineering, vol. 37, no. 2, pp. 247–262, 2011.*
- [8]: **Joseph B. Pine.** *Mass Customization: The New Frontier in Business Competition.* 1992.
- [9]: **F. Gottschalk, W. van der Aalst, M. Jansen-Vullers, and M. La Rosa,** *Configurable workflow models,* *IJCIS, vol. 17, pp. 177–221, 2008.*