Manuel d'utilisation de DesEsper

Stage - Analyse de logs applicatifs

INSA Lyon - Liris / EDF - R&D

Sébastien Gassmann

16 octobre 2008

Table des matières

1 Introduction

Ce manuel d'utilisation est basé sur la version 1.0 de DesEsper [Gas08] . Cette version s'utilise en ligne de commande avec deux fichiers de configuration : un pour la description des champs des évènements à analyser et l'autre pour les séquences.

2 Comment démarrer ?

2.1 Lancement du système DesEsper

L'outil DesEsper a été écrit en java, exécutable donc sur de multiples architectures et des systèmes d'exploitation différents. Son utilisation se fait en mode console, la configuration dans des fichiers XML. Les séquences à détecter et la description des champs des évènements se chargent au début du lancement de DesEsper. Une fois chargées, elles ne peuvent plus être modifiées sans arrêter le système DesEsper.

Exemple de commande d'exécution de DesEsper

```
java DesEsper -q sequences.xml -d descLogs.xml -s logs.txt -f >> sortie.txt
```

2.2 Description des paramètres

2.2.1 Le paramètre -q <fichier>

Le paramètre *-q* ("q" pour "queries") précède le nom du fichier contenant les séquences à détecter. Ce paramètre est obligatoire et doit forcément précéder un fichier xml de séquences.

2.2.2 Le paramètre -d <fichier>

Le paramètre *-d* ("d" pour "description") précède le nom du fichier contenant les descriptions des champs des évènements à analyser. Ce paramètre est obligatoire et doit forcément précéder un fichier xml de description de champs.

2.2.3 Le paramètre -s <fichier>

Le paramètre *-s* ("s" pour "source") précède le nom du fichier contenant le flux (fichier fini ou non fini). Ce paramètre est obligatoire et doit forcément précéder un fichier.

2.2.4 Le paramètre -f

la paramètre *-f* ("f" pour "file") précise si l'on veut que le système s'arrête à la fin du fichier source ou continue à attendre les nouveaux évènements. Ce paramètre est facultatif.

3 Création des fichiers de configuration

3.1 Le fichier « descLogs.xml »

Ce fichier xml est précédé de l'option *-d* et contient la description des événements contenus dans le flux.

La structure du fichier xml est constitué d'une section *evenements* (entre <evenements> et </evenements>) contenant les sections *description* et *groupes*.

La section *description* (balises <description> et </description>) contient les champs qui structurent un événement.

Un champ est caractérisé par les balises <champ> et </champ> contenant les attributs obligatoires *"type"*, *"debut"* et *"taille"* et l'attribut *"format"* nécessaire que si le type est *"Date"*.

L'attribut type contient le type du champ (*"String"* et *"Date"* et bientôt *"int"*).

L'attribut *"debut"* contient la position sur la ligne d'évènement où commence le champ.

L'attribut *"taille"* contient la valeur de la taille du champ, à partir de la position *"debut"*.

L'attribut *"format"* contient le format de détection de la date pour le champ *"Date"*.

Spécification de l'attribut *"format"*

Une date peut être composée de :
- Année : 2 ou 4 chiffres *"yy/YYYY"*
- Mois : 1 ou 2 chiffres *"mm"*
- Jour : 1 ou 2 chiffres *"dd"*
- Heure : 2 chiffres *"hh"*
- Minutes : 2 chiffres *"ii"*
- Secondes : 2 chiffres *"ss"*
- dixième : 1 chiffre *"x"*
- centième : 1 chiffre *"c"*
- millième : 1 chiffre *"l"*

Tous les caractères différents de « y, Y, d, h, i, s, x, c et l » seront ignorés.

La section *groupes* est composé d'un ou plusieurs groupes définis par les balises <groupe> et </groupe> contenant l'attribut *"nom"*. L'attribut *"nom"* contient le nom donné au groupe. Une étape de séquence portera sur un groupe. (Conférer 3.2 Le fichier "séquences.xml").

Chaque groupe contient zéro, un ou plusieurs critères de sélection. Ces critères sont définis de la même façon que les critères des étapes des séquences (voir ci-dessous).

Exemple de suite d'évènements et du contenu du fichier de description associé

Exemple de suite d'évènement : un fichier de logs

```
        Dis.G1  03/06   11H25'47,8"     MARCHE  POMPE   REGUL.
        Dis.G1  03/06   11H25'50,2"     FREINAGE EN      COURS
        APP.G2  03/06   11H36'05,8"     X.R.O   EXCITE
*ALARME APP.G1  03/06   11H36'05,9"     DEFAUT  VANNE   de TETE
        APP.G2  03/06   11H36'06,8"     MARCHE  POMPE   VANNE   de TETE
        APP.G2  03/06   11H36'07,1"     GROUPE   EN     MARCHE
```

Description des champs en XML

```
<evt>
 <description>
  <champ type="String" debut="0" taille="7">Alarme</champ>
  <champ type="String" debut="8" taille="8">TypeEvt</champ>
  <champ type="Date" debut="16" taille="19" format="dd:mm   hh:ii:ss">Date</champ>
  <champ type="String" debut="40" taille="0">Description</champ>
 </description>
 <groupes>
  <groupe nom="G1">
   <critere champ="TypeEvt">G1</critere>
  </groupe>
  <groupe nom="G2">
   <critere champ="TypeEvt">G2</critere>
  </groupe>
  <groupe nom="tous">
  </groupe>
 </groupes>
</evt>
```

3.2 Le fichier « sequences.xml »

Le fichier xml précédé de l'option *-q* contient la définition des séquences à détecter.

La structure du fichier se compose d'une grande section *sequences* contenant l'ensemble des séquences, une séquence est encadrée des balises <sequence> et </sequence> qui contiennent les attributs *"nom"* et *"unique"*. L'attribut *"nom"* contient le nom de la séquence, et *"unique"* la valeur booléenne (vrai ou faux) de l'unicité de la détection.

Définir l'unicité de la détection à "vrai" permet de spécifier si l'on souhaite qu'une séquence ne puissent être détecter qu'une fois pour un même évènement.

A l'intérieur des balises <sequence> et </sequence> est contenu les différentes étapes de la séquence, une étape représente un événement à détecter.

Une *étape* est caractérisée par les balises <etape> et </etape> contenant les attributs *"source"* et *"timeout"*, l'attribut *"source"* contient le nom du groupe de l'évènement à détecter et *"timeout"* la valeur en milliseconde du temps permis avant de détecter l'évènement suivant.

A l'intérieur des balises <etape> et </etape> est contenu l'ensemble des critères que doit satisfaire l'évènement pour être détecté. Le nombre de critère peut être nul et est illimité.

Un *critère* est caractérisé par les balises <critere> et </critere> contenant les attributs *"champ"*, *"comparateur"* et *"formatDate"*.

L'attribut *"champ"* est indispensable, il contient le nom du champ sur lequel le *critère* est appliqué, les attributs *"comparateur"* et *"formatDate"* sont nécessaires que si le champ est *"Date"* et que l'on souhaite effectuer une comparaison de date.

Les valeurs permises pour le comparateur sont « inf,sup, infegal,supegal,egal,diff », et pour *"formatDate"*, les mêmes restrictions que pour la description du champ *"Date"* dans le fichier de description des évènements.

Les balises <critere> et </critere> contiennent l'expression régulière à détecter ou la date (selon le format défini dans *"formatDate"*) à comparer. Les expressions régulières sont des expressions construites à partir de constantes et d'opérateurs, elles correspondent aux expression régulières utilisées par les classes Java "Pattern" et "Matcher". Pour consulter la liste des opérateurs et leurs utilisations, il faut se reporter à la référence [Cay].

Les balises <regle> et </regle> contiennent également un couple de balise <sortie> et </sortie>. Ces balises contiennent la description de la *sortie* désirée lorsqu'une séquence est complètement détectée. La *sortie* peut être formatée (saut de ligne, décalage de caractère, textes etc.)

Les termes suivants sont réservés et sont remplacés par leurs descriptions dans la sortie :

- **Premier** : affiche le premier événement de la séquence.
- **Dernier** : afffiche le dernier événement de la séquence.
- **NomSequence** : affiche le nom de la séquence.
- **DureeTotale** : Affiche la durée totale de la séquence (du premier événement au dernier)
- **NumeroSequence** : Affiche le numéro de la séquence (le numéro de la séquence correspond au nombre de fois que la séquence a été détectée)
- **evtX :N : :C :NomDuChamp :C :N :** : Affiche un champ particulier d'un évènement particulier :
  - **X** : Numéro de l'évènement

- **:C :NomDuChamp :C :** : Un ou plusieurs champs peuvent être spécifiés.
- **:C :Date :K :FormatDeLaDate :C :** : Pour le champ "Date", un format peut être spécifié, respectant le format de java : *HH :mm :ss dd/MM/yyyy*

- **:E : :E :** , le texte et les termes inscrits entre les balises :E : sont répétés pour chaque étape (événement)

Ces termes ne sont disponibles que pour chaque étape (entre les balises :E :)

- **:C :NomDuChamp :C :** : affiche la valeur du champ de l'évènement (champ défini dans la description des évènements plus les champs « Date » et « Log » prédéfini).
- **:C :Date :K :FormatDeLaDate :C :** : Pour le champ "Date", un format peut être spécifié, respectant le format de java : *HH :mm :ss dd/MM/yyyy*
- **DureePartiel** : Affiche la durée entre le premier événement et l'évènement en cours
- **DureeEtape** : Affiche la durée entre l'évènement précédent et l'évènement en cours
- **NumeroEtape** : Affiche le numéro de l'étape (position de l'étape dans la séquence)

Exemple de séquence et du fichier XML associé

Exemple de Séquence une erreur de connexion est défini par une séquence de 3 évènements séparé par un interval de temps de 2 secondes maximum : le premier évènement doit contenir "attempt 1/3", le deuxième évènement doit contenir "attempt 2/3" et le troisième "attempt 3/3".

Traduction de la séquence en XML

```
<sequences>
 <sequence nom="Afficher" unique="vrai">
  <etape groupe="tous" timeout="2000">
   <critere champ="description">attempt 1/3</critere>
  </etape>
  <etape groupe="tous" timeout="2000">
   <critere champ="description">attempt 2/3</critere>
  </etape>
  <etape groupe="tous" timeout="2000">
   <critere champ="description">attempt 3/3</critere>
  </etape>
  <sortie>NomSequence a durée DureeTotale
Voici les évènements qui sont apparus :
:E::C:Log:C::E:</sortie>
 </sequence>
</sequences>
```

3.3 Le fichier « Logs.txt »

Le fichier précédé de l'option *-s* est le fichier contenant le flux d'évènement. Le fichier peut être un flux continu, ou un fichier fini.

3.4 Le fichier « sortie.txt »

Dans l'exemple de commande donnée, la sortie est redirigée dans un fichier de sortie mais peut-être redirigée vers la sortie standard, vers un autre logiciel de traitement, vers un flux etc.

4 Exemples d'analyse

Le système DesEsper permet plusieurs possibilités d'analyse sur un flux : calcule de la durée entre deux évènements, détection de séquences, réorganisation chronologique, etc. Afin d'illustrer ces possibilités d'analyse, deux exemples seront développés dans cette section.

4.1 Premier exemple : Etude de la durée de démarrage d'une centrale hydraulique, logs fourni par EDF

4.1.1 Présentation des logs

Les logs sont issus de l'automate qui contrôle la centrale hydraulique du site de Sisteron. Ils contiennent des séquences de démarrage, d'arrêt et d'autre évènement de surveillance (Alarme, interventions, etc.).

Les logs sont enregistrés dans un fichier de 44800 évènements sur une période allant du 2 juin 2007 au 31 décembre 2007.

Extrait du fichier de logs

```
        Dis.G1  03/06   11H25'47,8"     MARCHE  POMPE   REGUL.
        Dis.G1  03/06   11H25'50,2"     FREINAGE EN      COURS
        APP.G2  03/06   11H36'05,8"     X.R.O   EXCITE
*ALARME APP.G1  03/06   11H36'05,9"     DEFAUT  VANNE   de TETE
        APP.G2  03/06   11H36'06,8"     MARCHE  POMPE   VANNE   de TETE
        APP.G2  03/06   11H36'07,1"     GROUPE   EN     MARCHE
```

4.1.2 Préparation du fichier de description des champs

Chaque ligne de logs peut être découpée en 4 champs positionnés toujours à la même position d'une ligne à l'autre :

- Un champ *Alarme* du debut de ligne au caractère 7.
- Un champ *TypeEvt* du caractère 8 au caractère 15.
- Un champ *Date* du caractère 16 au caractère 35 avec comme format : "dd :mm hh :ii :ss".
- Un champ *Description* du caractère 40 au caractère "fin de ligne" et de taille variable.

Dans cette analyse, nous étudierons les démarrages de deux groupes de la centrale : le groupe G1 et le groupe G2.

Afin de distinguer les évènements de ces deux groupes, et les évènements de disparition et d'apparition, quatres groupes seront crées avec comme critère la présence de "G1" ou de "G2" et "APP" ou "Dis" dans le champ *TypeEvt*.

Description des champs en XML

```
<evt>
 <description>
  <champ type="String" debut="0" taille="7">Alarme</champ>
  <champ type="String" debut="8" taille="8">TypeEvt</champ>
  <champ type="Date" debut="16" taille="19" format="dd:mm   hh:ii:ss:x">Date</champ>
  <champ type="String" debut="40" taille="0">Description</champ>
 </description>
 <groupes>
  <groupe nom="APP G1">
   <critere champ="TypeEvt">APP</critere>
   <critere champ="TypeEvt">G1</critere>
  </groupe>
  <groupe nom="DIS G1">
   <critere champ="TypeEvt">Dis</critere>
   <critere champ="TypeEvt">G1</critere>
  </groupe>
  <groupe nom="APP G2">
   <critere champ="TypeEvt">APP</critere>
   <critere champ="TypeEvt">G2</critere>
  </groupe>
  <groupe nom="DIS G2">
   <critere champ="TypeEvt">Dis</critere>
   <critere champ="TypeEvt">G2</critere>
  </groupe>
 </groupes>
</evt>
```

4.1.3 Préparation du fichier xml de séquences

On souhaite étudier les durées par étape des démarrage des deux groupes G1 et G2 de la centrale. Un démarrage du groupe G1 est caractérisé par la séquence suivante :

```
APP.G1 X.R.O EXCITE
APP.G1 GROUPE EN MARCHE
APP.G1 MARCHE POMPE REGUL.
APP.G1 FREINAGE EN COURS
Dis.G1 VERROU ENGAGE
Dis.G1 FREINAGE EN COURS
APP.G1 DISJ. EXCITA FERME
APP.G1 VANNE de TETE OUVERT.
APP.G1 DISJ. FERME
```

Chaque évènement est représenté par une étape de la séquence. Afin de d'écrire l'évenment, un seul critère suffit, il suffit de mettre le contenu attendu du champ *Description"*.

Traduction en XML

```
<sequences>
 <sequence nom="Demarrage G1" unique="vrai">
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">X.R.O EXCITE</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">GROUPE EN MARCHE</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">MARCHE POMPE REGUL</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">FREINAGE EN COURS</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">VERROU ENGAGE</critere>
  </etape>
  <etape groupe="DIS G1" timeout="1200000">
   <critere champ="Description">FREINAGE EN COURS</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">DISJ EXCITA FERME</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">VANNE TETE OUVERT</critere>
  </etape>
  <etape groupe="APP G1" timeout="1200000">
   <critere champ="Description">DISJ FERME</critere>
  </etape>
 </sequence>
</sequences>
```

Le timeout entre chaque étape a été fixé à 1200000 millisecondes, temps maximum observé en général.

La séquence pour le groupe G2 est la même, il suffit de remplacer "G1" par "G2".

Afin de pouvoir utiliser les résulats, le format de sortie sera le format CSV, ainsi la sortie sera facilement importable dans un tableur ou un outil d'analyse statistique.

Sortie en XML

```
<sortie>NomSequence;:E:DureeEtape;:E:
</sortie>
```

4.1.4 Exécution de l'exemple

L'exemple étudié ici porte sur un fichier fini de logs. L'option *-f* sera donc utilisée pour que l'analyse s'arrête à la fin du fichier.

Ligne de commande à exécuter Contenu des fichier DesEsper-exemple1.sh (à exécuter sous Linux) ou DesEsper-exemple1.bat (à exécuter sou Windows) :

```
java DesEsper -q sequence-exemple1.xml -d desc-exemple1.xml -s logs-edf.txt -f >> resultats_exemple1.csv
```

Le résultats est visualisable simplement en ouvrant le fichier resultats-exemple1.csv crée ou en l'important dans un logiciel d'analyse.

4.2 Deuxième exemple : Etude sur un flux continu généré aléatoirement, message provenant d'un distributeur de billet.

4.2.1 Présentation des évènements

Les évènements à analyser sont générés aléatoirement et simulent le fonctionnement basique d'un distributeur de billet. Voici les différents messages qui peuvent être générés :

- "Erreur code"
- "Carte refusé"
- "Code accepté"
- "Retrait"
- "Consultation solde"
- "Erreur inconnue"
- "Retrait refusé"
- "Déconnexion"

4.2.2 Préparation du fichier de description des champs

Exemple d'évènement

```
26     2008-08-03 15:35:10:266 Message : Consultation solde
27     2008-08-03 15:35:11:242 Message : Consultation solde
28     2008-08-03 15:35:12:108 Message : Carte refusé
29     2008-08-03 15:35:12:524 Message : Retrait refusé
30     2008-08-03 15:35:13:461 Message : Consultation solde
31     2008-08-03 15:35:14:034 Message : Carte refusé
32     2008-08-03 15:35:14:084 Message : Retrait refusé
33     2008-08-03 15:35:14:189 Message : Erreur code
34     2008-08-03 15:35:14:512 Message : Carte refusé
35     2008-08-03 15:35:14:805 Message : Code accepté
36     2008-08-03 15:35:15:758 Message : Erreur code
37     2008-08-03 15:35:16:115 Message : Retrait
```

Chaque ligne de logs peut être découpée en 4 champs positionnés toujours à la même position d'une ligne à l'autre :

- Un champ *Numero* du debut de ligne au caractère 6.
- Un champ *Date* du caractère 7 au caractère 30 avec comme format : "YYYY-mm-dd hh :ii :ss :xcl".
- Un champ *Message* du caractère 32 au caractère "fin de ligne" et de taille variable.

Description des champs en XML

```
<evt>
 <description>
  <champ type="String" debut="0" taille="6">Numero</champ>
  <champ type="Date" debut="7" taille="23" format="YYYY-mm-dd hh:ii:ss:xcl">Date</champ>
  <champ type="String" debut="40" taille="0">Message</champ>
 </description>
 <groupes>
  <groupe nom="Tous">
  </groupe>
 </groupes>
</evt>
```

On remarque qu'il n'ya pas eu le besoin de créer de groupes, donc un seul groupe "Tous" sans critère est spécifié. Tous les évènements arrivant appartiendront à ce groupe.

4.2.3 Préparation du fichier xml de séquences

Plusieurs séquences vont être définies afin d'établir une surveillance du distributeur.

Première séquence : détecter une suite de 3 évènements "Erreur code" (500 millisecondes entre chaque évènement) et envoyer le message "3 erreurs : carte avalée".

Deuxième séquence : détecter une suite de 2 évènements "Erreur code" suivi de l'évènement "Code accepté" et envoyer le message "Attention la prochaine fois".

Troisième séquence : détecter une suite de 3 séquences "Code accepté" suivi de l'évènement "Deconnexion" et envoyer le message "Abus" et retourner les numéro des évènements.

Quatrième séquence : détecter une suite de 3 évènements "Erreur inconnue" et envoyer le message "Anomalie".

Traduction en XML

```
<sequences>
 <sequence nom="Première séquence" unique="vrai">
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Erreur code</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Erreur code</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Erreur code</critere>
  </etape>
  <sortie>carte avalée</sortie>
 </sequence>
 <sequence nom="Deuxième séquence" unique="vrai">
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Erreur code</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Erreur code</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Code accepté</critere>
  </etape>
  <sortie>Attention la prochaine fois</sortie>
 </sequence>
 <sequence nom="Troisième séquence" unique="vrai">
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Code accepté</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Deconnexion</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Code accepté</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Deconnexion</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Code accepté</critere>
  </etape>
  <etape groupe="Tous" timeout="5000">
   <critere champ="Message">Deconnexion</critere>
  </etape>
  <sortie>Abus : Evt :E::C:Numero:C:
:E:</sortie>
 </sequence>
 <sequence nom="Quatrième séquence" unique="vrai">
```

```
  <etape groupe="Tous" timeout="1000">
   <critere champ="Message">Erreur inconnue</critere>
  </etape>
  <etape groupe="Tous" timeout="1000">
   <critere champ="Message">Erreur inconnue</critere>
  </etape>
  <etape groupe="Tous" timeout="1000">
   <critere champ="Message">Erreur inconnue</critere>
  </etape>
  <sortie>3 "erreur inconnue" : Anomalie</sortie>
 </sequence>
</sequences>
```

4.2.4 Exécution de l'exemple

L'exemple étudié ici porte sur un fichier continu d'évènement. Il faudra lancer le générateur d'évènement et rediriger sa sortie vers un fichier "flux.txt" qui sera analysé par le système DesEsper.

Ligne de commande à exécuter Fichier DesEsper-exemple2.sh ou DesEsper-exemple2.bat

Il faut tout d'abord lancer le générateur d'évènement avec cette commande : Fichier EvtGenerator.sh ou EvtGenerator.bat (Vérifier que le fichier flux.txt n'éxiste pas)

```
java EvtGenerator >> flux.txt
```

Et lancer le système DesEsper en parallèle : Fichier DesEsper-exemple2.sh ou DesEsper-exemple2.bat

```
java DesEsper -q sequence-exemple2.xml -d desc-exemple2.xml -s flux.txt -f
```

Le générateur d'évènement peut être arrêté et relancé à volonté sans stopper DesEsper, le système DesEsper attendra l'arrivée de nouveaux évènements.

Références

[Cay] Bernard Caylux. "expressions régulières". `http://prevert.upmf-grenoble.fr/Prog/Java/CoursJava/expressionsRegulieres.html`.

[Gas08] Sébastien Gassmann. "desesper", Juillet 2008.