Analyse stochastique de séquences d'événements discrets pour la découverte de signatures

Philippe Bouché, Marc Le Goc

LSIS, UMR CNRS 6168, Domaine Universitaire St Jérôme,
13397 Marseille cedex 20, France
philippe.bouche@lsis.org; marc.legoc@lsis.org

Résumé. Cet article concerne la découverte de signatures (ou modèles de chroniques) à partir d'une séquence d'événements discrets (alarmes) générée par un agent cognitif de surveillance (Monitoring Cognitive Agent ou MCA). Considérant un couple (Processus, MCA) comme un générateur stochastique d'événements discrets, deux représentations complémentaires permettent de caractériser les propriétés stochastiques et temporelles d'un tel générateur : une chaîne de Markov à temps continu et une superposition de processus de Poisson. L'étude de ces deux représentations duales permet de découvrir des « signatures » décrivant les relations stochastiques et temporelles entre événements dans une séquence. Ces signatures peuvent alors être utilisées pour reconnaître des comportements spécifiques, comme le montre l'application de l'approche à un outil de production industriel piloté par un système Sachem, le MCA développé et utilisé par le groupe Arcelor pour aider au pilotage de ses outils de production.

1 Introduction

Nos travaux concernent les systèmes à base de connaissances de surveillance des processus dynamiques, appelés « Monitoring Cognitive Agent » par la suite (figure 1). Ces systèmes décrivent les évolutions du processus surveillé au moyen d'événements qui, selon le contexte, peuvent être qualifiés d'alarmes (ou d'avertissements) et adressés à l'Opérateur (Le Goc et Frydman, 2004). Les événements sont produits suivant un principe de discrétisation spatiale qui consiste à positionner le niveau d'un signal dans un ensemble d'intervalles de valeurs ou plages. Un événement est généré lorsqu'un signal entre dans une nouvelle plage.

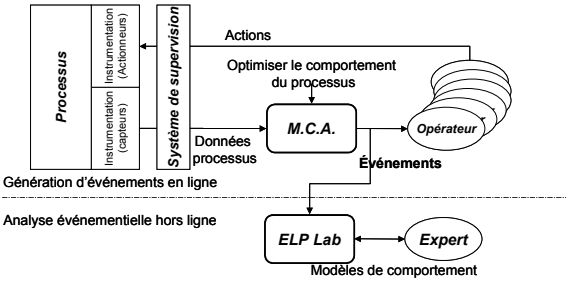

FIG. 1- *Monitoring Cognitive Agent*

Les séquences d'événements générées selon ce principe sont porteuses d'information sur l'évolution du processus surveillé. Notre objectif consiste à définir une méthode et un environnement logiciel, le laboratoire ELP, pour aider à extraire cette information et la représenter sous la forme de connaissances directement interprétables par un Expert. Ce type de connaissances est en effet utile à l'activité de diagnostic des défauts de comportement du processus surveillé (Dousson, 1999).

Notre approche du diagnostique est centrée sur la reconnaissance de séquences particulières d'événements, signe de comportements problématiques. Une séquence problématique est spécifiée sous la forme de « signature », un réseau où les nœuds sont des types d'événements et les liens des contraintes temporelles sur les événements. Le langage ELP (Event Language for Process) a été conçu pour représenter ce type de connaissance (Frydman et al, 2001).

La question adressée dans ce papier est la suivante : étant donné un ensemble de comportements problématiques observés dans une série d'expériences et les séquences d'événements associées, quelles sont les signatures qui caractérisent au mieux ces comportements ? Des questions similaires sont traitées dans le domaine du Data Mining avec des algorithmes tels que PASCAL (Mannila, 1995). Dans un article récent, Manilla dresse un bilan des limites de ces algorithmes et en identifie le principal défaut : les relations entre événements que ces algorithmes permettent de découvrir sont trop locales pour constituer une véritable représentation des séquences (Mannilla, 2002). Il invite donc à rechercher des algorithmes adoptant un point de vue plus global sur la recherche de signatures et propose d'investiguer la théorie des chaînes de Markov.

Nos travaux s'inscrivent dans cette perspective. Ils reposent sur l'idée que, sous certaines conditions, le processus de génération des alarmes peut être considéré comme un générateur $SDEG_{(Pr, MCA)}$ stochastique d'événement discret qui peut être modélisé sous les formes duales d'une superposition de processus de Poisson composés et d'une chaîne de Markov homogène à temps discret.

La section suivante définit la notion de signature puis introduit le modèle $SDEG_{(Pr, MCA)}$. La section 3 décrit la démarche d'analyse d'une séquence d'événements discrets et l'algorithme « BJT » conçu pour la découverte de signatures. Une application de cet algorithme à un processus industriel surveillé par un système Sachem est proposée en section 4. La conclusion de l'article propose un premier bilan de ces travaux.

2 Modélisation

2.1 Notion de signature

Un ensemble 'Ψ_x' de n seuils $(\psi_k)_{k=1\ldots n}$ $\psi_k \in \Re$, pour une variable 'x' définit un ensemble de n+1 plages de valeurs (intervalles) noté $R_x = \{r_l\}_{l=0,\ldots,n}$ tel que $r_0=[-\infty, \psi_1[$, $r_l=[\psi_l, \psi_{l+1}[$ et, $r_n=[\psi_n, +\infty[$. Un événement discret 'e_k' est un triplet (t_k, x, i) où $t_k \in \Gamma=\{t_i\}_{ti\in\Re}$ est la date de l'événement, 'x' est le nom de la variable associée à 'e_k' dont l'évolution temporelle est décrite par une fonction '$x(t)$' (i.e. un signal) de la variable réelle et 'i' est l'indice d'un intervalle dans 'R_x' ($i \in \aleph$).

Un événement 'e_k' est généré à chaque fois que '$x(t)$' entre dans une nouvelle plage :

$$\forall t_k \in \Re, \forall r_i \in R_x, \exists t_{k-1} < t_k, \quad x(t_{k-1}) \notin r_i \wedge x(t_k) \in r_i \Rightarrow e_k \equiv (t_k, x, i) \quad (1)$$

En dénotant par ' E ' l'ensemble des événements, une fonction notée 'd' fournit la date d'un événement :

$$d : E \to \Gamma, \quad \forall e_k \equiv (t_k, x, i) \in E, d(e_k) = t_k \quad (2)$$

Le type 'T_i' d'un événement '$e_k \equiv (t_k, x, i)$' est le couple (x, i) et est dénoté : '$e_k ::T_i$'. Une "signature" est un ensemble de relations binaires entre des types d'événements de la forme $R(e_i::T_i, e_o::T_o, [\tau^-,\tau^+])$ telle que :

$$\forall T_o, T_i \in T, \forall [\tau^-, \tau^+] \in \Re^2, \forall e_n, e_k \in E, \quad (e_n :: T_o) \wedge (e_k :: T_i) \wedge (d(e_n) - d(e_n) \in [\tau^-, \tau^+])$$
$$\Rightarrow R(e_n :: T_o, e_k :: T_i, [\tau^-, \tau^+]) \quad (3)$$

où '$e_i::T_i$' (resp. '$e_o::T_o$') dénote d'un événement d'entrée de type 'T_i' (resp. de sortie de type 'T_o'), $d(e_i) < d(e_o)$, et '$[\tau^-,\tau^+]$' est la fenêtre temporelle pour observer l'occurrence d'un événement du type de sortie '$e_o::T_o$' après l'occurrence de l'événement du type d'entrée '$e_i::T_i$' de sorte que : $(d(e_o)- d(e_i)) \in [\tau^-,\tau^+]$.

Une telle relation peut être décomposée en 2 relations :

- Une relation séquentielle $R_S(e_i::T_i, e_o::T_o)$ qui est une relation orientée d'un type d'événement d'entrée vers type d'événement de sortie :
$$\forall T_o, T_i \in T, \exists e_k, e_n \in E, (e_n :: T_o) \wedge (e_k :: T_i) \wedge (d(e_k) \le d(e_n)) \Rightarrow R_S(e_n :: T_o, e_k :: T_i) \quad (4)$$
- Une contrainte temporelle $R_T(d(e_i::T_i), d(e_o::T_o), [\tau^-,\tau^+])$ où '$d(e_k::T_k)$' dénote la date de l'occurrence d'un evt de type 'T_k' :
$$\forall d(e_i :: T_i), d(e_o :: T_o) \in \Gamma, d(e_o :: T_o) - d(e_i :: T_i) \in [\tau^-, \tau^+]$$
$$\Rightarrow R_T(d(e_i :: T_i), d(e_o :: T_o), [\tau^-, \tau^+]) \quad (5)$$

Les éléments présentés dans cette section sont développés dans (Le Goc, 2005).Cette décomposition met en lumière les deux aspects complémentaires d'une signature. L'idée est alors d'exploiter cette complémentarité pour faciliter l'étude des propriétés liant les événements discrets dans une séquence.

2.2 Processus de Poisson

Le principe de discrétisation spatiale possède des similarités avec l'approche dite « POT » (Peak Over Thresholds) qui montre que, sous certaines conditions relativement faciles à vérifier en pratique, la couple (Processus, MCA) peut être considéré comme une superposition de processus de Poisson (Lang et al., 1999).

Considérant une séquence $\omega=(e_k)_{k=1\ldots n}$ d'événements discrets de la forme $e_k=(t_k, x, i)$, les durées entre 2 événements successifs constituent un ensemble de variables aléatoires τ_1, τ_2, τ_3, … telles que : $\forall\ k\ge1$, $P[\tau_k > 0]=1$. Le processus de comptage des événements de ω est le processus n(t) tel que : $\forall t_{k-1}, t_k, n(t_k)=n(t_{k-1})+1$ avec $n(0)=0$. Un processus de dénombrement est le processus aléatoire $(N(t); t \ge 0)$ tel que :

$$\forall t \ge 0, N(t) = \max\{n(t) \ge 0 : S_k \le t\} \quad (6)$$

où $S_k = \sum_{j=1}^{k} \tau_j$ $\forall k \ge 1$ avec $S_0=0$. La durée écoulée entre deux événements successifs e_{k-1} et e_k dans ω est donnée par : $\tau_k = S_k - S_{k-1}$. Elle est appelée « durée de vie » d'un état du processus $(N(t); t \ge 0)$. Lorsque les durées de vies τ_k sont des variables aléatoires indépendantes et identiquement distribuées, le processus $(N(t); t \ge 0)$ est un processus de renouvellement. Lorsque les durées de vies sont distribuées selon une loi exponentielle de la forme $\lambda e^{-\lambda t}$, λ étant une constante réelle positive, le processus de renouvellement est un

processus de Poisson homogène d'intensité λ. La probabilité d'occurrence du prochain événement est alors donnée par :

$$P[N(t) = 1] = \lambda t\, e^{-\lambda t} \quad (7)$$

L'intensité λ correspond à la moyenne des durées de vie du processus de Poisson. La loi exponentielle est dite "sans mémoire" au sens où l'espérance mathématique du temps d'attente du prochain évènement est une constante :

$$E(\tau) = E(\tau - t > \tau > t) = \frac{1}{\lambda} (8)$$

2.3 Superposition et composition de processus de Poisson

Considérons un processus de Poisson (N(t) ; t≥0) d'intensité λ qui dénombre les événements dans une séquence $\omega=(e_k ::T_i)_{k=1\ldots n}$ contenant m types d'événements (m>0) tel que, en notant p_i la probabilité qu'une occurrence d'événement soit de type i :

- $\forall i \in \aleph$, $0 < i \leq m$, $0 < p_i < 1$,
- $p_1 + p_2 + \ldots + p_m = 1$

Lorsque les attributions de type sont indépendantes les unes des autres, un processus de Poisson d'intensité λ_i, ($N_i(t)$; $t \geq 0$), peut être définit pour chacun des types évènements dont au moins une occurrence est présente dans ω. Dans ce cas, le processus de Poisson (N(t); $t \geq 0$) est une superposition de m processus de Poisson indépendants ($N_i(t)$; $t \geq 0$). De la même manière, le processus de comptage des « transitions » entre événements de types T_i et T_j est une superposition de mx(m-1) processus de Poisson ($N_{ij}(t)$; t≥0) d'intensité λ_{ij}. Elle peut être représentée par une matrice $Q=[q_{ij}]$ contenant les propriétés temporelles de paires d'événements successifs de types T_i et T_j :

$$q_{ij} = \lambda_{ij}, \quad q_{ii} = -\sum_{j \neq i} \lambda_{ij} \quad et \quad \sum_j q_{ij} = 0 \quad (9)$$

Enfin, un processus aléatoire (X(t) ; $t \geq 0$), avec $X(t) = \sum_{j=1}^{N(t)} Y_j$ est un processus de Poisson composé d'intensité λ et de distribution de gain F lorsque :

- (N(t) ; $t \geq 0$) est un processus de Poisson d'intensité λ
- $Y_1, Y_2, Y_3, \ldots$ sont des variables aléatoires indépendantes et identiquement distribuées selon F,
- la suite aléatoire ($Y_1, Y_2, Y_3, \ldots$) et le processus (N(t); $t \geq 0$) sont indépendants.

Ainsi, 2 processus de Poisson ($N_i(t)$; t≥0) et ($N_j(t)$; t≥0) dénombrant respectivement les occurrences d'événements de type T_i et T_j mutuellement indépendants, donnent naissance à 2 de processus de Poisson composés d'intensité $\lambda_i^j=\lambda_i$ et $\lambda_j^i=\lambda_j$, l'un ayant pour gain les durées écoulées entre 2 occurrences successives de types T_i et T_j, l'autre ayant pour gain les durées écoulées entre 2 occurrences successives de types T_j et T_i. Ces gains sont distribués selon une loi exponentielle de paramètre λ_i^j et λ_j^i.

Par conséquent, m processus de Poisson ($N_i(t)$; t≥0) dénombrant les occurrences d'événements de type T_i mutuellement indépendants, donnent naissance à m*m-1 processus de Poisson composés d'intensité $\lambda_i^j=\lambda_i$, $i \neq j$, $i=1\ldots m$, $j=1\ldots m$, ayant pour gain les durées écoulées entre deux occurrences de types d'événement T_i et T_j distribuées selon une loi exponentielle de paramètre λ_i^j.

2.4 Chaînes de Markov

Soit un espace d'état $S=\{i\}_{i=0,\,1,\,\ldots,m}$, $m\in\aleph$, $m>0$. Une chaîne de Markov à temps continu $\mathbf{X}=(x_k;\ k>0)$ est une suite de variable aléatoire de S tel que $\forall k\in\aleph$, $k\geq 0$, et $\forall i_0,\ i_1,\ \ldots,\ i_k\in S$:

$$P\left[x(t_k)=i_k \mid x(t_{k-1})=i_{k-1}, x(t_{k-2})=i_{k-2}, \cdots, x(t_0)=i_0\right]= P\left[x(t_k)=i_k \mid x(t_{k-1})=i_{k-1}\right] \quad (10)$$

Cette condition est appelée 'propriété de Markov' : la probabilité d'une transition d'un état 'i' à un état 'j' ne dépend que du fait que la chaîne soit dans l'état 'i'. En particulier, cela ne dépend pas du temps passé dans l'état 'i'. Cette propriété correspond à la propriété d'absence de mémoire de la distribution exponentielle. La théorie des processus de Poisson est ainsi liée à celle des chaînes de Markov (Cassandras, 2001).

Le passage de l'un à l'autre de ces modèles s'effectue en considérant les événements dénombrés par un processus de Poisson comme marquant l'occurrence d'états dans une chaîne de Markov. Appliqués aux événements générés par discrétisation spatiale, l'espace d'état S est confondu avec l'ensemble T des types d'événements : une transition d'état dans une chaîne de Markov ($x(t_{k-1})=i$, $x(t_k)=j$) correspond à une séquence binaire $\omega_{ij}=(e_{k-1}::T_i,\ e_k::T_j)\subseteq\omega$. Lorsque le processus de comptage ($N(t);\ t\geq 0$) est un processus de Poisson homogène, la chaîne de Markov correspondante est homogène : $\forall i,\ j\in S$, et $\forall k\in\aleph$, $k>0$, la probabilité conditionnelle $P[x(t_k)=j \mid x(t_{k-1})=i]$ ne dépend pas de k. La chaîne de Markov est alors entièrement définie par sa matrice des taux de transition $Q=[q_{ij}]$ où 'q_{ij}' représente le taux de Poisson des transitions de l'état 'i' vers l'état 'j' et correspondent aux taux des processus de Poisson ($N_{ij}(t)$; $t\geq 0$) d'intensité λ_{ij} dans une séquence ω (équation 9). Les probabilités de transition d'états sont alors données par la matrice des probabilités de transition $P=[p_{ij}]$:

$$\forall j\neq i,\quad p_{ij}=\frac{q_{ij}}{-q_{ii}}\quad et\quad p_{ii}=0 \quad (11)$$

La chaîne de Markov étant homogène, la matrice P peut être estimée avec la méthode du 'maximum de vraisemblance' (équation 12), comme dans le cas discret, mais à la différence que dans une chaîne de Markov à temps continu, il ne peut y avoir de transition d'un état sur lui même ($p_{ii}=0$). Pour maintenir la dualité de la double représentation, il est préférable d'associer une chaîne de Markov à temps discret à la superposition de processus de Poisson, cette dernière étant un modèle à temps continu.

$$p_{ij}=\frac{q_{ij}}{-q_{ii}}=\frac{\lambda_{ij}}{\sum_j \lambda_{ij}}=\frac{\dfrac{\sum_l x_{ij}(t_l)}{T}}{\sum_j \dfrac{\sum_l x_{ij}(t_l)}{T}}=\frac{\sum_l x_{ij}(t_l)}{\sum_j \sum_l x_{ij}(t_l)} \quad (12)$$

Ainsi, à une séquence $\omega=(e_k::T_k)_{k=0,1,\ \ldots,\ m}$, contenant m+1 événements de n types, il est possible d'associer une chaîne de Markov à temps discret de n états $X=\{x_i\}_{i=1\ldots n}$, et n*n transitions $X_{ij}=\{x_{ij}\}_{i=1..n,\ j=1\ldots n,}$. La matrice P est constante et donnée dans l'équation 12 où '$x_{ij}(t)$' représente l'occurrence d'une transition de l'état i à l'état j de la chaîne de Markov à l'instant t. Cette matrice contient les propriétés stochastiques des transitions d'état associées aux successions d'occurrence de types d'événements dans la séquence. Les propriétés

temporelles des transitions d'état sont représentées par une superposition de processus de Poisson composés (paragraphe 2.3).

En synthèse, lorsque le dénombrement des événements générés par un couple (Pr, MCA) peut être considéré comme une superposition de processus de Poisson, le couple (Pr, MCA) peut être modélisé sous la forme d'un générateur d'événement discret $SDEG_{(Pr, MCA)}$ dont les propriétés stochastiques sont portées par la matrice P des probabilités de transition d'état d'une chaîne de Markov homogène à temps discret, les propriétés temporelles étant représentées par une superposition de processus de Poisson composés. Ces deux représentations duales peuvent être exploitées séparément pour découvrir des signatures de types d'événements (équation 3).

3 Algorithme BJT

3.1 Principes de base

Le rôle de l'algorithme BJT est d'aider à découvrir les signatures d'un type d'événement dans une séquence $\omega=(e_k::T_k)_{k=0,1, \ldots, m}$, contenant m+1 événements de n types. Cette recherche est effectuée en 3 étapes :

- 1°) Représentation des relations séquentielles sous la forme d'un arbre des transitions les plus probables. Cette étape transforme la matrice P en un arbre dont les nœuds sont des types d'événements et les arcs des relations orientées d'un type d'événement T_i vers un type d'événement successeur T_j.
- 2°) Estimation des contraintes temporelles. Cette étape consiste à apposer des étiquettes sur les arcs contenant une mesure des durées moyennes entre occurrence de type T_i et occurrence de type T_j (λ_i^j).

La troisième étape, actuellement effectuée en dehors de l'algorithme BJT, permet d'identifier les signatures comme une branche de l'arbre des transitions les plus probables à partir d'un test d'anticipation. Ce test est basé sur le calcul du ratio entre le nombre de sous-séquences respectant les contraintes séquentielles et temporelles d'une branche complète de l'arbre et le nombre de sous-séquences respectant les contraintes de la même branche privée du type d'événement racine (i.e. le type final). Une branche sera considérée comme une signature potentielle si son taux d'anticipation est supérieur ou égal à 50 %.

Une fois une signature identifiée, la séquence ω peut être représentée dans une forme simplifiée appelée « séquence-modèle » ω_S associée à une signature S. Dans une séquence-modèle, les événements d'un type T_i se produisent tous les $1/\lambda_i$. Cette représentation facilite l'interprétation des signatures produites à l'aide de l'algorithme BJT.

3.2 Identification des relations séquentielles

Le nombre de transitions d'un état 'i' vers un état 'j' dans une chaîne de Markov homogène correspond au nombre d'occurrences de la sous séquence $(e_{k-1}::T_i, e_k::T_j)$ dans une séquence ω de m+1 événements (et inversement). Trouver l'événement de type 'T_i' qui précède le plus souvent un événement de type 'T_j' dans une séquence ω correspond donc à trouver l'état 'i' dans la chaîne de Markov associée qui précède l'état 'j' avec la plus grande probabilité. Le même raisonnement peut être fait pour l'état 'i', c'est à dire pour le type 'T_i'.

La matrice P est utilisée pour trouver le chemin le plus probable menant à un événement de type 'T_j'. Cette matrice est pondérée par la probabilité de l'occurrence d'une séquence ω_{ij} du type ($e_{k-1}::T_i$, $e_k::T_j$) afin de minimiser le problème de la dispersion des λ_{ij} correspondant à une sur-représentation de séquences ω_{ij} rares dans ω :

$$P(\omega_{ij}) = \frac{\sum_l \omega_{ij}(t_l)}{m} = \frac{\sum_l x_{ij}(t_l)}{\sum_i \sum_j \sum_l x_{ij}(t_l)} \quad (13)$$

Le principe de l'algorithme consiste à transformer la matrice BJ=[b_{ij}]=[p_{ii}*P(ω_{ij})] en un arbre dont la racine est l'événement de type 'T_j' et les branches représentent la suite des transitions dans la chaîne de Markov menant à cette racine. En notant 'i_o' l'état pour lequel on cherche le chemin, n la profondeur de l'arbre, $\delta(j)$ l'ensemble des prédécesseurs les plus probables de l'état j, $\Sigma(k)$ l'ensemble des états du chemin qui doit être trouvé à l'étape k, l'algorithme est le suivant :

```
//Initialisation
1. C = ∅, Σ(0)={i_o}
//Boucle sur la profondeur de l'arbre
2. ∀k=1,…, n
    3. ∀j∈Σ(k-1)
        4. δ(j)←{i | b_ij∈MAX_i(b_ij) ∧ x_ij∉C}
        5. ∀i∈δ(j), C←C∪x_ij
        6. Σ(k)←δ(j)
```

FIG. 2 – *Algorithme de transformation de BJ en arbre*

La largeur de l'arbre dépend de la manière dont la fonction 'MAX' est codée. Dans notre cas, l'arbre est développé afin d'avoir un nombre paramétrable de branches à chaque noeud.

3.3 Identification des relations temporelles

La seconde phase de l'algorithme 'BJT' exploite la représentation d'une séquence sous la forme d'une superposition de processus de Poisson composés pour déterminer les relations temporelles entre paires de types d'événements.

L'algorithme utilise n*(n-1) automates indépendants (figure 3), chacun étant consacré à la capture des transitions entre deux types d'événements T_i et T_j. L'algorithme compte le nombre de transitions entre deux occurrences successives de type T_i et T_j, n_{ij}, dans une séquence et cumule les durées s_{ij} entre les événements correspondants. Ces calculs sont effectués à chacune des transitions d'un état à l'autre.

La moyenne des temps entre deux occurrences d'événements de type 'T_i' et 'T_j' est simplement donnée par le rapport « s_{ij}/n_{ij} ». Cette moyenne est alors portée comme étiquette d'une relation séquentielle entre les types 'T_i' et 'T_j' sur l'arbre des relations séquentielles.

Une fois l'arbre des transitions les plus probables construit et complété par les relations temporelles, le test d'anticipation permet de ne retenir que les branches susceptibles d'être des signatures. Lorsqu'une signature a été identifiée, la séquence ω peut être représentée sous sa forme simplifiée de « séquence-modèle ».

// Boucle sur les événements de ω
1. Lire (e_k::T_i)
// i est l'index de T_i dans T_ω
// Mémorisation de la date de l'occurrence
2. $d_i = d(e_k)$
// Pour chaque automate concerné par l
// Execution des fonctions de calcul
3. $\forall j \neq i$, si Q_{ij}=1 alors
    // Calcul du nombre de transition
    4. $n_{ji} = n_{ji} + 1$
    // Calcul du temps total des transitions
    5. $s_{ji} = s_{ji} + d(e_k) - d_j$
    // Transition d'état
    6. $Q_{ji} = 1$, $Q_{ij} = 0$

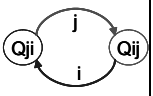

FIG. 3 – *Algorithme d'estimation des relations temporelles*

3.4 Séquence-modèle d'une séquence associée à une signature

Dans une séquence-modèle ω_S d'une séquence ω possédant k types d'événements associée à une signature S, les événements sont répétés et distribués temporellement selon les intensités λ_i des processus de Poisson (i.e. pour chaque type T_i, les durées inter-événements sont constantes et égales à $1/\lambda_i$).

Pour un type T_i, la date de la $k^{iéme}$ occurrence d'événements de type T_i est donnée par : $d(e_k ::T_i) = k*1/\lambda_i$. Les occurrences sont ensuite ordonnées pour produire la séquence-modèle ω_S. Ainsi, par construction, ω_S possède une période, calculée en cherchant un réel n unique tel que : $\exists n \in \Re$, $\forall i \in \aleph$, $\exists m_i \in \aleph$, $\lambda_i * n = m_i$. L'entier m_i est le nombre d'occurrence du type T_i au cours de la période n. Cela signifie que, '$e_k ::T_i$' étant le $k^{ième}$ événement de la période de la séquence ω_S, l'occurrence de date $(j*n)+d(e_k ::T_i)$ de ω_S est aussi de type T_i :

$$\forall e_k \in \omega_S, \forall j \in \aleph, e_k ::T_i \Rightarrow \exists e' \in \omega_S, e'::T_i \wedge d(e')=(j*n)+d(e_k) \quad (14)$$

Cette propriété permet de représenter la séquence modèle ω_S sur une période, la suite étant obtenue par répétition. Une séquence modèle ω_S est une représentation approchée de la séquence ω correspondant à une signature S produite par l'aide de l'algorithme BJT. Les différences observées entre ω et ω_S proviennent de la modélisation par processus de Poisson, notamment au niveau de l'estimation des taux de Poisson λ_i par le maximum de vraisemblance. Mais les résultats produits par ce modèle sont suffisamment pertinents pour être acceptés par des experts, ce qu'illustre la section suivante.

4 Application

L'application proposée dans cette section concerne un outil de production industriel[1] piloté à l'aide d'un système Sachem, le système de diagnostic à base de connaissance temps réel développé par le groupe Arcelor, premier producteur mondial d'acier. Sachem a été initialement conçu pour aider au pilotage de haut-fourneau. Parce qu'il est générique (Le Goc and Gaéta, 2003), Sachem a été décliné sur d'autres outils comme celui qui fait l'objet

[1] Nous ne pouvons pas décrire le processus étudié pour des raisons de confidentialité

de cette étude. Les systèmes Sachem sont basés sur le principe de la discrétisation spatiale (Le Goc, 2005). Les types d'événements sont désignés par un nombre entier appartenant à l'intervalle [1000, 3000].

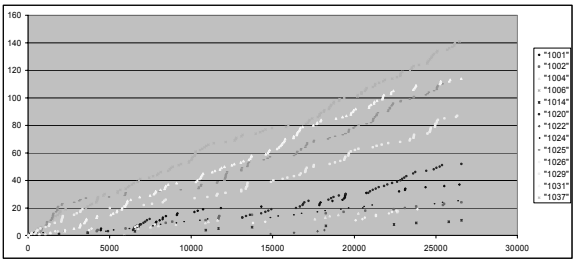

FIG. 4 - *Courbes de Poisson associé à la séquence ω*

La séquence ω analysée couvre 19 jours du 25/04/2003 au 13/05/2003 où le processus de dénombrement des événements de ω se comporte comme un processus de Poisson d'intensité λ=1,4 événement/heure, soit un événement toutes les 40 minutes environ (figure 4). Le type d'événement dont la signature est recherchée, 1026, correspond à un enjeu en matière d'optimisation de la consommation énergétique.

	1001	1002	1004	1006	1014	1020	1022	1024	1025	1026	1029	1031	1037
1001	6	1	0	15	0	2	0	0	6	0	4	4	0
1002	3	1	1	4	10	0	0	0	0	0	2	3	0
1004	2	4	0	2	0	2	0	0	3	0	1	3	1
1006	11	5	7	35	1	15	1	6	18	5	19	23	0
1014	0	1	0	4	0	0	0	0	1	4	0	0	0
1020	1	2	2	18	0	5	0	1	6	3	5	10	0
1022	0	0	0	1	0	0	0	1	0	0	2	0	0
1024	1	0	2	3	0	3	0	0	2	1	6	7	0
1025	2	4	3	20	0	7	2	6	26	3	25	12	1
1026	0	0	0	9	0	4	0	1	3	0	0	2	0
1029	4	1	2	13	0	7	0	3	30	3	14	13	0
1031	8	5	2	21	0	6	1	7	19	1	11	36	2
1037	0	0	0	0	0	1	0	0	0	0	2	1	0

TAB 1 - *Matrice de dénombrements des transitions dans ω*

	1001	1002	1004	1006	1014	1020	1022	1024	1025	1026	1029	1031	1037
1001	14,665	0,4074	0	91,657	0	1,6295	0	0	14,665	0	6,5178	6,5178	0
1002	5,805	0,645	0,645	10,32	64,499	0	0	0	0	0	2,58	5,805	0
1004	3,44	13,76	0	3,44	0	3,44	0	0	7,7399	0	0,86	7,7399	0,86
1006	12,829	2,6507	5,1953	129,88	0,106	23,856	0,106	3,817	34,353	2,6507	38,276	56,088	0
1014	0	1,548	0	24,768	0	0	0	0	1,548	24,768	0	0	0
1020	0,2921	1,1683	1,1683	94,632	0	7,3018	0	0,2921	10,515	2,6287	7,3018	29,207	0
1022	0	0	0	3,87	0	0	0	3,87	0	0	15,48	0	0
1024	0,6192	0	2,4768	5,5728	0	5,5728	0	0	2,4768	0,6192	22,291	30,341	0
1025	0,5578	2,2313	1,2551	55,783	0	6,8335	0,5578	5,0205	94,274	1,2551	87,161	20,082	0,1395
1026	0	0	0	65,993	0	13,036	0	0,8147	7,3326	0	0	3,2589	0
1029	2,752	0,172	0,688	29,068	0	8,4279	0	1,548	154,8	1,548	33,712	29,068	0
1031	8,3253	3,2521	0,5203	57,367	0	4,683	0,1301	6,3741	46,96	0,1301	15,74	168,59	0,5203
1037	0	0	0	0	0	3,87	0	0	0	0	15,48	3,87	0

TAB 2 - *Matrice BJ de la séquence ω*

La séquence ω contient 646 événements de 13 types et concerne 11 variables. La chaîne de Markov associée comporte donc 13 états et 156 transitions potentielles. Le tableau 1 présente la matrice de dénombrement des transitions d'états pour la séquence ω, le tableau 2 la matrice « BJ » (par souci de lisibilité, toutes les valeurs b_{ij} ont été multipliées par 10 000).

L'application de l'algorithme 'BJT' à la séquence ω produit l'arbre de la figure 5. La première branche de cet arbre peut être retenue comme une signature du type d'évènement 1026 car elle possède un taux d'anticipation de plus de 63% : sur les 11 sous-séquences de ω respectant la signature tronquée de son nœud final (i.e. la branche {1004, 1031} -> 1002 -> 1014}, 7 respectent la signature complète.

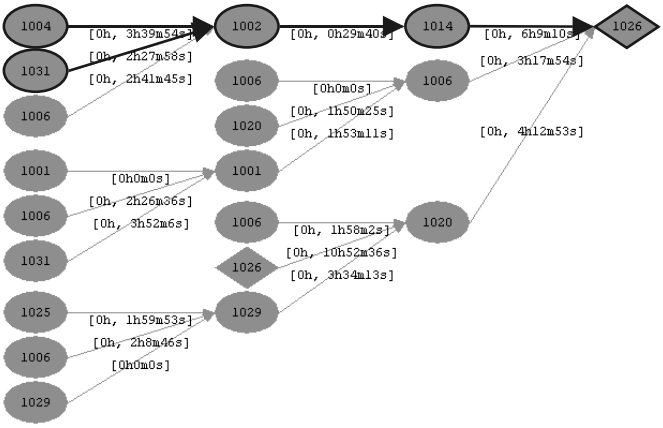

FIG. 5 - *Arbre du type d'événement '1026'*

De plus, cette signature « explique » 7 des 17 occurrences d'événements 1026, c'est-à-dire 41% des occurrences. Cette mesure est appelée « taux de couverture » d'une signature. Elle signifie qu'en l'occurrence, au moins 2 autres signatures sont nécessaires pour décrire les liens entre les événements de type 1026 et les autres types d'événements dans 'ω'.

evt	1006	1031	1029	1020	1001	1002	1026	1004	1014
Lambdas	7,89	6,69	5,85	4,3	2,03	1,08	1,21	0,84	0,48
Nombre	789	669	585	430	203	108	121	84	48

TAB 3 - *Intensités des processus de Poisson*

La séquence-modèle ω_{1026} de ω pour la signature du type 1026 est calculée à partir des intensités des processus de Poisson mesurés sur la séquence ω (Tableau 3). La période de la séquence modèle ω_{1026} est de 100 jours et possède 3037 événements. L'application de l'algorithme BJT sur cette séquence conduit par construction à la même signature pour le type 1026.

ω_{1026} débute de la manière suivante (les noms des variables ont été omis) : {(1006, 0.1267); (1031, 0.1494); (1029, 0.1709); (1020, 0.2325); (1006, 0.2538); (1031, 0.2988); (1029, 0.3418); (1006, 0.3801); (1031, 0.4482); (1020, 0.465); (1001, 0.4926); (1006,

0.5068); (1029, 0.5127); (1031, 0.5976); (1006, 0.6335); (1029, 0.6836); (1020, 0.6975); (1031, 0.747); (1006, 0.7602); (1026, 0.8403); (1029, 0.8545); (1006, 0.8869); (1031, 0.8964); (1002, 0.9259); … }.

Ces résultats ont été présentés et validés par un expert en conduite du processus étudié. Des résultats similaires ont été obtenus sur un haut-fourneau piloté par Sachem. Ils montrent que l'approche proposée produit des résultats opérationnels sur une application industrielle.

5 Conclusion

L'approche proposée dans cet article considère un couple (processus, MCA) comme un générateur $SDEG_{(Pr,\ MCA)}$ stochastique d'événements discrets typés qui peut être modélisé sous les formes duales d'une superposition de processus de Poisson composés et d'une chaîne de Markov homogène à temps discret.

L'intérêt de cette approche réside dans la séparation des études des propriétés séquentielles et temporelles d'une séquence d'événements produit par un $SDEG_{(Pr,\ MCA)}$: les relations séquentielles entre événements sont tirées de l'exploitation des propriétés stochastiques de la chaîne de Markov alors que les relations temporelles sont estimées à partir des propriétés temporelles des processus de Poisson.

L'algorithme BJT présenté dans cet article exploite cette dualité d'une part en transformant la matrice des probabilités de transition dans une chaîne de Markov en un arbre décrivant les différents chemins à probabilité maximale liant les types d'événements entre eux, et d'autre part, en complétant cet arbre par les contraintes temporelles tirées de la représentation par processus de Poisson. Un simple test d'anticipation permet de sélectionner dans cet arbre les branches susceptibles d'être des signatures d'un type d'événement. Cette approche a été validée sur un processus industriel surveillé par un système Sachem.

L'algorithme BJT est un des outils développés et intégrés au sein du « laboratoire ELP », un environnement Java dédié à l'analyse des séquences d'événements discrets générés par les systèmes Sachem.

Références

Aycard O., Laroche P., Charpillet F. (1998), Mobile Robot Localization in Dynamic Environments using Place Recognition, the 1998 IEEE International Conference on Robotics and Automation (ICRA'98), pages 3135-3140.

Cassandras, C. G., and S. Lafortune (2001), Introduction to dicrete event systems, Kluwer Academic Publishers.

Cocozza-Thivent C. (1997), Processus stochastiques et fiabilité des systèmes, Edition SPRINGER.

Dousson C. and Vu Duong T (1999), Discovering Chronicles with Numerical Time Constraints from Alarms Logs for Monitoring Dynamic Systems, the 1-th International Join conference on Artificial Intelligence (IJCAI'99), pp. 620-626.

Frydman C., Le Goc M., Torres L. and Giambiasi N (2001), Knowledge-Based diagnosis in SACHEM using DEVS models, Special Issues of Transaction of Society for Modeling and Simulation International (SCS) on Recent Advances in DEVS Methodology, Tag Gon Kim Ed., Vol. 18, N°3, p. 147-158.

Lang M., Ouarda T.B.M.J. and Bobée B (1999), Towards operational guidelines for over-threshold modelling, Journal of hydrology 225, p103-117, Edition ELSEVIER.

Le Goc M. and Gaéta M. (2003). Modeling Structures in Generic Space, a Condition for Adaptiveness of Monitoring Cognitive Agent. Journal of Intelligent and Robotic Systems - Theory and Applications, Kluwer Academic Publishers, 2003

Le Goc M. and Frydman C. (2004), The Discrete Event Concept as a Paradigm for the "Perception Based Diagnosis" of Sachem, Journal of Intelligent and Robotics Systems, Kluwer Academic Publishers.

Le Goc M. (2005). Sachem, a Real Time Intelligent Diagnosis System based on the Discrete Event Paradigm. To appear in Simulation: Transaction of the Society for Modelling and Simulation International.

Mannila H., Toivonen H. and Verkamo I. (1995), Discovering Frequent Episodes in Sequences, First International conference on Knowledge Discovery and Data Mining (KDD'95), pp. 210-215.

Mannila H. (2002), Local and Global Methods in Data Mining: Basic Techniques and Open Problems, 29th International Colloquium on Automata, Languages and Programming, Volume 2380, pp. 57-68.

Rabiner L. R. (1989), A tutorial on Hidden Markov Models and selected applications in Speech Recognition, Fellow, IEEE, 77(2), P257-289.

Reinert, G., Schbath, S. (1998), Compound Poisson and Poisson process approximations for occurrences of multiple words in Markov chains, Journal of Computational Biology, Volume 5, Number 2.

Summary

This paper aims at showing a method to the discovery of signature (or model of chronicles) starting from a discrete event sequence (alarms) generated by a Monitoring Cognitive Agent ("MCA" in the continuation). Regarding a couple (process, MCA) as a stochastic generator of discrete events, two complementary representations make it possible to characterize the stochastic and temporal properties of such a generator: a chain of Markov at continuous time and a superposition of processes of Poisson. The study of these two dual representations makes it possible to discover "signatures" describing the stochastic and temporal relations between events in a sequence. These signatures can then be used to recognize specific behaviours, as the application of the approach shows it to an industrial production equipment controlled by a Sachem system, the MCA used by the Arcelor group to help the monitoring of its production equipments.